

THESIS / THÈSE

DOCTEUR EN SCIENCES

Techniques de visualisation d'information pour l'analyse de situations coopératives

Otjacques, Benoît

Award date:
2009

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix
Institut d'informatique
Namur (Belgique)



Techniques de visualisation d'information pour l'analyse de situations coopératives

Dissertation présentée par Benoît Otjacques
en vue de l'obtention du grade
de Docteur en Sciences (orientation : informatique)
à l'Institut d'Informatique
des Facultés Universitaires Notre-Dame de la Paix – Namur

Composition du jury :

Prof. Jean-Marie Jacquet (FUNDP, Namur, président)
Dr Thomas Baudel (ILOG S.A., Paris)
Prof. Vincent Englebert (FUNDP, Namur)
Prof. Jean-Luc Hainaut (FUNDP, Namur)
Prof. Pierre Leclercq (Université de Liège)
Prof. Monique Noirhomme (FUNDP, Namur, promoteur)
Prof. Gilles Venturini (Université de Tours)

Défense privée : 10 octobre 2008

Défense publique : 6 janvier 2009



Cette thèse est dédiée à mon épouse Valérie et à ma fille Hélène

Chapitre 1 : Introduction

« Il n'est pas de vent favorable pour celui qui ne sait où il va. »

Sénèque

1. Remerciements

Une entreprise d'une certaine ampleur peut rarement être accomplie seul. A l'issue de ces années de recherche, la portée de cette observation m'apparaît encore plus générale que je ne l'imaginais.

L'heure de la gratitude a sonné et certains noms m'apparaissent incontournables.

A titre individuel, je citerai en premier lieu la Professeure Monique Noirhomme qui a supervisé mes travaux. Je lui exprime ma gratitude d'avoir accepté ma candidature en thèse et le sujet que je proposais. Les résultats présentés dans ce manuscrit n'auraient pu être atteints sans son aide. En particulier, sa grande expertise en méthodologie d'évaluation et en statistiques a permis d'obtenir des conclusions qui peuvent être qualifiées d'objectives. Les relations amicales dans la forme mais strictes sur le fond que nous avons construites au cours de ces années de réflexion commune restent à mes yeux un modèle de coopération fructueuse.

A titre personnel, je souhaite aussi remercier Fernand Feltz, directeur scientifique du département 'Informatique, Systèmes et Collaboration' du Centre de Recherche Public – Gabriel Lippmann où je suis employé. Sa double compétence de scientifique et de manager force mon respect depuis plus de dix ans.

Plusieurs articles scientifiques ont été publiés pour diffuser les résultats de mes travaux. Je témoigne ici de ma gratitude pour les co-auteurs de ceux-ci.

Divers chercheurs m'ont aidé à concrétiser mes idées théoriques et à les implémenter sous la forme de prototypes logiciels. Je les remercie pour leur aide à ce niveau.

Les évaluations de logiciels n'auraient pu être menées à bien sans la participation des personnes qui jouèrent le rôle de sujets de test. Elles méritent elles aussi mes remerciements.

Je tiens également à faire preuve de gratitude envers les membres de mon jury de thèse pour les remarques pertinentes qu'ils ont formulées à propos de ce manuscrit ainsi que pour le débat enrichissant qu'ils ont suscité lors de la défense de cette thèse.

Pour conclure le volet scientifique, je voudrais aussi remercier d'une manière générale tous ceux qui, consciemment ou inconsciemment, ont contribué aux processus d'émergence, d'amélioration, de validation ou de réfutation des nombreuses idées qui ont émergé au cours de mes travaux.

A titre privé, je rend enfin hommage à mon épouse Valérie qui dut à de nombreuses reprises accepter les contraintes liées à la réalisation de mes travaux de thèse. Enfin, bien qu'elle ne soit pas encore capable d'en prendre la mesure complète, ma fille Hélène a illuminé de son sourire, de sa joie de vivre et de ses remarques déroutantes les années consacrées à cette thèse.

2. Motivation des travaux



La plupart des grandes entreprises humaines furent réalisées en associant les compétences individuelles et en mutualisant les ressources. A titre d'exemples, l'architecture mais aussi l'histoire des techniques regorgent de réalisations dans laquelle un nombre considérable de personnes agissent pour atteindre un but commun. Bien que les modalités d'exécution aient fondamentalement changé, le concept même de travail coopératif est donc probablement aussi ancien que la civilisation humaine.

L'informatique, bien que nettement plus récente, a réussi la gageure de percoler en quelques décennies dans toutes les disciplines scientifiques et d'influencer un nombre sans cesse croissant d'activités humaines. Il apparaît donc assez naturel que certains scientifiques aient étudié comment les systèmes informatiques pouvaient contribuer à la gestion d'activités de nature coopérative. Cette discipline, appelée '*Computer – Supported Cooperative Work*' a émergé dans les années 1980.

Alors que l'homme utilise depuis des siècles des représentations graphiques pour imaginer, concevoir ou modéliser des objets concrets tels que des outils ou des bâtiments, leur utilisation pour visualiser des éléments abstraits ne date que de la fin du 18^{ème} siècle. L'analyse scientifique de cette approche est actuellement connue sous le terme de '*visualisation d'information*'. Bien qu'elle soit très récente, cette nouvelle discipline se développe actuellement à un rythme exponentiel car elle tente de répondre à un des défis majeurs de la société de l'information : « Comment permettre à un être humain d'appréhender la sémantique contenue dans des données volumineuses, complexes et dynamiques ? »

Notre thèse se positionne comme une recherche multidisciplinaire à l'intersection de ces trois thématiques. Elle est consacrée à l'étude des techniques de visualisation d'information en vue d'améliorer la compréhension, l'animation, la gestion ou la supervision de plates-formes électroniques de support à des activités coopératives. Afin de prendre en compte toute la diversité de cette thématique de recherche, nous devons faire appel à des connaissances dans les domaines aussi divers que la visualisation d'information, la psychologie cognitive, le génie logiciel, le travail coopératif, le calcul numérique, les statistiques et la méthodologie d'évaluation de logiciels.



3. Démarche globale

Nous présentons dans cette section la démarche suivie au cours de cette thèse afin de mettre en évidence la cohérence globale de nos travaux.

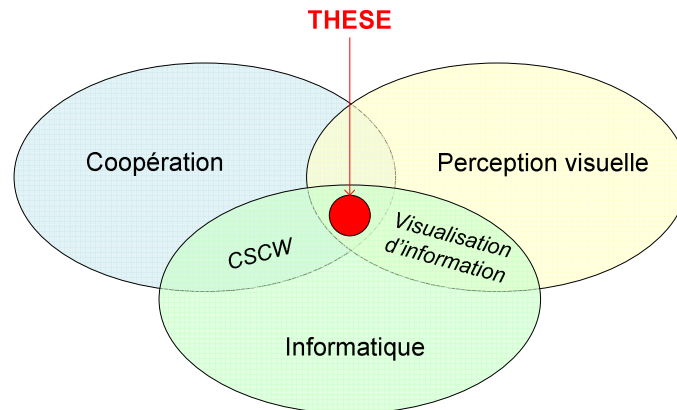


Fig. 1: Disciplines scientifiques sous-jacentes à la thèse

En premier lieu, il est essentiel de prendre conscience du caractère multidisciplinaire de notre approche. Nous ferons appel à trois grandes disciplines scientifiques : l'informatique, l'analyse de la coopération et l'étude de la perception visuelle. Comme le montre la Fig. 1, notre thèse se situe à l'intersection entre l'étude du travail de groupe supporté par l'informatique (*Computer Supported Cooperative Work – CSCW*) et la visualisation d'information (c.-à-d. la représentation graphique de données abstraites au moyen de logiciels).

La Fig. 2 illustre le cheminement que nous avons suivi et identifie les thématiques que nous avons étudiées.

L'informatique sert de fondation à l'ensemble de nos réflexions. En effet, d'une part, nous nous focaliserons sur un type particulier de situations coopératives dans lesquelles les outils informatiques jouent un rôle clé. D'autre part, nos travaux se concentrent sur l'étude des *logiciels* permettant de visualiser des données. A titre d'exemples, la sociologie de la coopération ou l'étude des mécanismes physiologiques permettant la vision se positionnent en dehors du champ de nos travaux.

La première étape de notre démarche a consisté à construire un état de l'art dans trois domaines précis : la coopération supportée par des outils électroniques, la visualisation d'information (et notamment les taxonomies proposées pour structurer cette discipline) et les techniques de visualisation utilisées pour représenter des données relatives à une situation de coopération.

Ensuite, afin de disposer d'un cadre de référence pour nos réflexions, nous avons conçu un modèle générique de la coopération (cf. Fig. 10, p. 33). Puis, nous avons étudié en détail les données qu'il était utile de visualiser pour aider à analyser une situation de coopération. Nos travaux nous ont amenés à nous intéresser en particulier au concept de conscience de groupe. Dans ce contexte, nous avons proposé une approche conceptuelle qui différencie les informations à visualiser selon leur anonymité et leur granularité. Sur la base de cette classification, nous avons identifié deux cas pertinents en pratique : la conscience de groupe individuelle (*Workspace Individual Awareness*) et la conscience de groupe globale (*Workspace Group Awareness*).

L'étude de cette dernière nous a conduit à aborder la question des métriques des activités coopératives. Comment résumer dans un indicateur global la complexité des interactions sous-tendant des activités coopératives ? Alors que la plupart des travaux antérieurs prenaient en compte la quantité d'interactions entre les personnes et éventuellement les ressources pour

évaluer le niveau de coopération, nous avons innové en proposant de nouveaux indices qui intègrent, dans une certaine mesure, la qualité de celles-ci en termes de motivation à coopérer.

Assez logiquement, l'étape suivante a visé à représenter graphiquement les indices précédemment définis pour supporter la conscience de groupe globale (WGA). L'utilitaire logiciel '*Ambient Flags*' s'inscrit dans cette démarche. En complément, nous avons également imaginé le logiciel de visualisation '*Clock View*' qui permet de supporter la conscience de groupe individuelle (WIA).

Notre étude de l'état de l'art sur les techniques de visualisation de données associées à un contexte coopératif a mis en évidence le besoin d'une meilleure structuration de ce domaine au niveau conceptuel. Nous avons donc entrepris l'élaboration d'une taxonomie que nous avons nommée IVF. Elle identifie un ensemble de propriétés permettant de décrire une situation coopérative, les aspects de celle-ci qu'il peut être utile de visualiser et les techniques utilisées pour représenter graphiquement ces derniers. Ces travaux théoriques nous ont aidé à concevoir différentes techniques originales de visualisation.

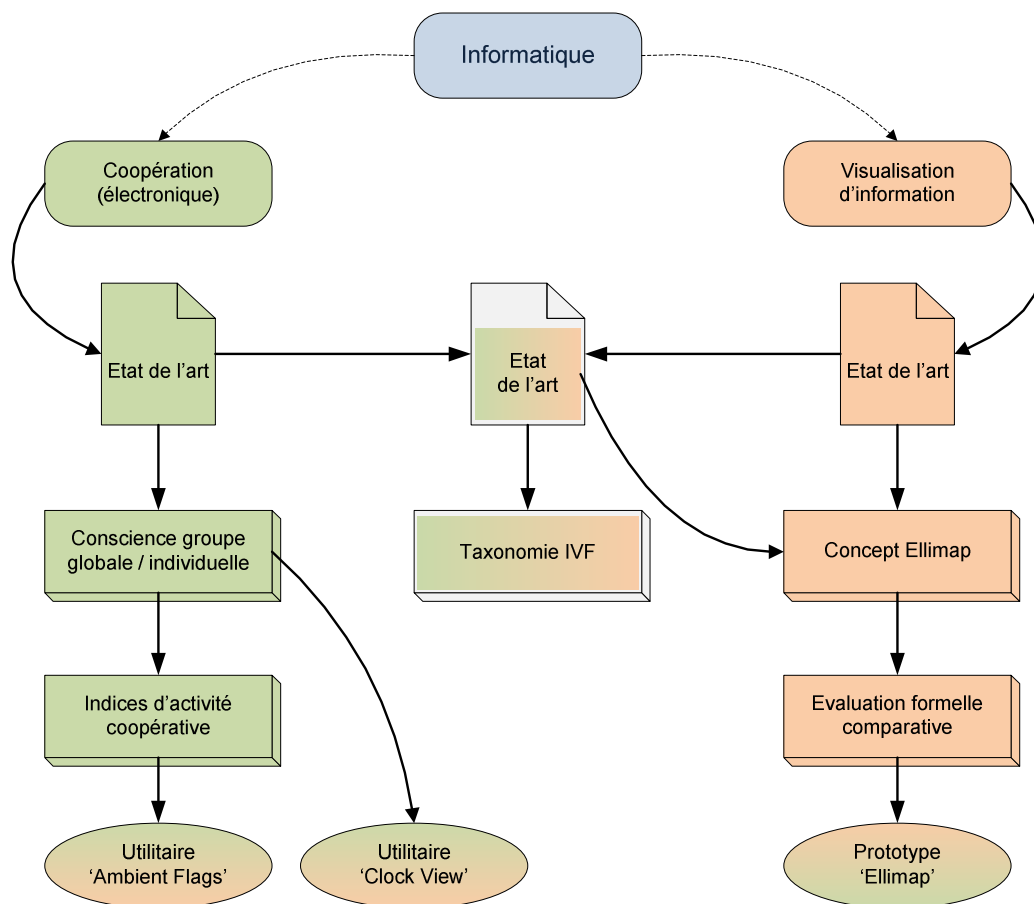


Fig. 2: Plan général des travaux de thèse

Constatant que dans un nombre significatif de cas, les données que nous cherchions à visualiser se présentaient sous la forme de structures hiérarchiques, nous avons ensuite entrepris l'étude des travaux antérieurs en matière de visualisation de données de ce type.

L'étude de la théorie '*Gestalt*' issue de la psychologie cognitive nous a inspiré et nous avons voulu intégrer ses lois dans notre démarche de conception d'une nouvelle technique de visualisation d'information. Dans un premier temps, nous avons confronté la technique des '*treemaps*', connue pour son approche originale de représentation de hiérarchies, avec la théorie '*Gestalt*'. Cette confrontation, assez inhabituelle, nous a incité à proposer une nouvelle approche, nommée '*ellimaps*', qui nous semblait plus conforme aux principes de la

théorie ‘*Gestalt*’. Cette nouvelle technique de visualisation fit l’objet diverses évolutions afin de résoudre ou d’atténuer ses faiblesses.

Au cours du processus de conception et de développement du prototype logiciel qui implémentait le concept d’*ellimaps*, plusieurs évaluations formatives furent réalisées afin de confronter nos résultats partiels avec des utilisateurs potentiels.

Nous avons également décidé de comparer formellement l’approche *ellimaps* avec la technique des ‘*treemaps*’ dont elle est conceptuellement issue. Comme nous souhaitons suivre une démarche méthodologie rigoureuse, nous nous sommes de nouveau tourné vers la littérature et nous avons étudié en détail la problématique de l’évaluation de logiciels et plus spécifiquement celle d’interfaces homme-machine. Sur cette base, nous avons identifié les biais potentiels dans la conception et la mise en œuvre d’une telle étude comparative. Pour répondre à certains d’entre eux, nous avons conçu et développé un environnement logiciel spécifique (appelé FESTIV) qui nous a permis de contrôler précisément les variables indépendantes de notre évaluation et de collecter des mesures les plus précises et significatives possibles.

Nous avons ensuite confronté notre intuition initiale et les résultats expérimentaux pour déterminer notamment si, dans les conditions expérimentales de notre étude, les *ellimaps* améliorent la perception de la structure des données par rapport aux *treemaps*.

Pour terminer, il est important de mentionner que nous avons utilisé nos différents prototypes pour visualiser des données issues de situations réelles de coopération. Différents exemples concrets sont présentés dans ce manuscrit.

4. Contributions majeures de la thèse

Les contributions de notre thèse sont résumées ci-dessous, en mettant l’accent sur les résultats les plus marquants qui furent publiés¹ dans des journaux scientifiques ou des conférences internationales. Nous les présenterons en nous référant à la démarche générale que nous avons suivie et qui est décrite dans la section précédente.

Une première contribution consiste dans notre réflexion sur la conscience de groupe d’un espace de travail qui nous a conduit à distinguer le caractère global ou individuel des informations communiquées à un utilisateur (cf. Chapitre 6 : Indices d’activité et Interactions, p. 150). Ensuite, nous nous sommes focalisés sur la question des métriques de l’activité coopérative qui s’avère indiscociable de la problématique de la conscience de groupe globale. Observant que la majorité des travaux antérieurs se sont limités à des métriques basées sur le volet quantitatif des interactions collaboratives, nous avons mis en évidence l’importance de l’aspect qualitatif. En termes de support à la coopération et de motivation des acteurs, il est trop simpliste de considérer que toutes les interactions possèdent le même poids. Nous avons donc conçu des indices de la coopération qui intègrent, grâce à une classification automatisable, la nature ‘coopérative’ des interactions entre les entités concernées (cf. Chapitre 6 : Indices d’activité et Interactions, p. 153).

Nos travaux sur ce thème ont fait l’objet des publications suivantes :

- ☑ Otjacques Benoît, Noirhomme Monique, Figueiredo José et Feltz Fernand (2006), *Composite Indexes as Metrics of Cooperative Activity*, 11^{ème} Conférence de l’Association Information & Management (AIM 2006), 7-9 Juin 2006, Luxembourg.
- ☑ Otjacques Benoît, Noirhomme Monique, Gobert Xavier et Feltz Fernand (2006), *Cooperation Indexes to Support Workplace Awareness*, 12th International Workshop on Groupware (CRIWG 2006), 17-21 Septembre 2006, Valladolid, Espagne, publié dans Lecture Notes in Computer Science, Springer.

¹ Toutes les publications mentionnées dans cette section ont fait l’objet d’une procédure d’évaluation (*review process*)

Ces articles concernaient essentiellement les concepts de conscience de groupe individuelle et globale d'une part et les indices d'activité coopérative d'autre part.

Notre étude des taxonomies de techniques de visualisation représente une seconde contribution importante de nos travaux. Nous avons analysé une quinzaine de taxonomies antérieures et nous avons identifié les éléments conceptuels qui les différencient (cf. Chapitre 3 : Visualisation d'information, p. 75). Sur cette base, nous avons identifié l'absence et le besoin d'une taxonomie spécifique portant sur les techniques de visualisation d'information relatives à des activités coopératives. Nous avons donc conçu une telle taxonomie (que nous avons appelée IVF, cf. Chapitre 5 : Taxonomie des techniques de visualisations de la coopération, p. 125).

Une première version de cette taxonomie IVF originale fut présentée lors de la conférence CDVE 2005. Une version plus détaillée fut décrite dans un article publié dans le journal scientifique ITCon.

- ☑ Otjacques Benoît, Noirhomme Monique et Fernand Feltz (2005), *Visualization of Interactions within a Project: the IVF Framework*, 2nd International Conference on Cooperative Design, Visualization and Engineering (CDVE 2005), 18-21 Septembre 2005, Mallorca, Espagne, publié dans Lecture Notes in Computer Science, Springer.
- ☑ Otjacques Benoît, Noirhomme Monique et Feltz Fernand (2006), *Taxonomy of the Visualization Techniques of Project-related Interactions*, Electronic Journal of Information Technology in Construction (ITCon), Vol. 11, Special issue on Process Modelling, Process Management and Collaboration, disponible à l'adresse <http://www.itcon.org/2006/41>.

Au point de vue des techniques de visualisation d'information proprement dites, la technique de visualisation *ellimaps* est un des résultats les plus importants de nos travaux (cf. Chapitre 7 : Ellimaps et dérivés, p. 189). Elle dérive de la technique des '*treemaps*' mais utilise des formes curvilignes (ellipses) plutôt que des rectangles. Cette modification, en apparence mineure, nécessite de nombreux développements tant au niveau géométrique qu'algorithmique pour obtenir un résultat final acceptable. Nous avons étudié en détail les avantages mais aussi les effets de bords indésirables consécutifs à la représentation de données hiérarchiques par le biais d'ellipses successivement incluses les unes dans les autres. Dans ce cadre, il faut mentionner notre algorithme 'DGC' (Déformations Géométriques Combinées) qui a constitué une évolution significative du concept initial des ellimaps car il permet de mieux exploiter l'espace d'affichage disponible (cf. Chapitre 7 : Ellimaps et dérivés, p. 202).

Nous pensons également avoir contribué à la réflexion sur les méthodologies d'évaluation d'interfaces homme-machine de logiciels, en particulier dans le cas de techniques de visualisation d'information. Notre approche basée sur l'identification et le contrôle strict des variables indépendantes considérées dans le plan d'expérience fut matérialisée dans le logiciel de test FESTIV (cf. Chapitre 8 : Ellimaps : Evaluation, p. 274). A notre connaissance, un seul logiciel véritablement comparable (HVTE) existait au moment où FESTIV fut implémenté. Ce dernier fut utilisé pour mettre en œuvre notre étude comparative formelle entre les *ellimaps* et les '*treemaps*'.

La description conceptuelle des *ellimaps* et les conclusions de notre étude comparative avec la technique des '*treemaps*' fut publiée au cours des journées EGC 2007 et, dans une version étendue, dans la Revue d'Intelligence Artificielle' (RIA) en 2008.

- ☑ Otjacques Benoît, Collin Pierre, Feltz Fernand et Noirhomme Monique (2007), *Ellimaps : une nouvelle technique de visualisation pour améliorer la perception de la structure de données arborescentes*, 7^{èmes} journées francophones « Extraction et Gestion des Connaissances » (EGC'07), Atelier sur la visualisation et l'extraction de connaissances, 23-26 Janvier 2007, Namur.

- ☑ Otjacques Benoît, Collin Pierre, Feltz Fernand et Noirhomme Monique (2008), *Ellimaps : une technique basée sur la loi d'inclusion pour représenter des hiérarchies avec nœuds pondérés*, Revue d'Intelligence Artificielle (RIA), Vol. 22, N° 3-4, 2008.

La technique des *ellimaps* fut initialement imaginée pour représenter graphiquement des informations permettant l'analyse de plates-formes électroniques de coopération (cf. Chapitre 9 : Prototype 'Ellimaps', p. 305). Son utilisation dans un tel contexte fut détaillée dans deux articles présentés dans des conférences internationales.

- ☑ Otjacques Benoît, Noirhomme Monique, Gobert Xavier, Collin Pierre et Feltz Fernand (2007), *Visualizing the activity of a web-based collaborative platform*, 11th International Conference on Information Visualization (IV'07), 4-6 Juillet 2007, Zurich, Suisse, publié dans IEEE Computer Society Editions.
- ☑ Otjacques Benoît, Noirhomme Monique et Feltz Fernand (2007), *Innovative visualization to monitor scientific cooperative activities*, 4th International Conference on Cooperative Design, Visualization and Engineering (CDVE 2007), 16-20 Septembre 2007, Shanghai, Chine, publié dans Lecture Notes in Computer Science, Springer.

Enfin, parmi les évolutions de la technique des *ellimaps* figure également le concept d'*ellimaps* réservoirs qui a pour but de visualiser les niveaux de consommation de ressources associées aux nœuds d'une hiérarchie (cf. Chapitre 7 : Ellimaps et dérivés, p. 222). Un dernier article portant spécifiquement sur les *ellimaps* réservoirs fut présenté au cours de la conférence ICDIM 2008. Outre son intérêt immédiat, l'approche développée dans les *ellimaps* réservoirs met également en évidence le caractère relatif de l'hypothèse habituelle selon laquelle l'espace libre entre les formes géométriques successivement incluses les unes dans les autres doit être considéré de l'espace perdu. En effet, cet espace peut s'avérer très utile, par exemple, pour afficher des informations complémentaires sur les nœuds de la hiérarchie à visualiser (étiquettes, niveaux de consommation).

- ☑ Otjacques Benoît, Collin Pierre, Feltz Fernand et Noirhomme Monique (2008), *Visualizing consumption in weighted hierarchies*, 3rd IEEE International Conference on Digital Information Management (ICDIM'08), 13-16 Novembre 2008, Londres, Royaume-Uni, publié dans IEEE Computer Society Editions.

5. Plan

Cette thèse est organisée en dix chapitres.

Le premier rassemble les éléments d'usage dans une introduction : remerciements, motivation générale des travaux, plan de la thèse, contributions majeures des travaux et liste des publications scientifiques associées.

Dans toute démarche scientifique, un état de l'art s'impose comme une des tâches initiales. Nous avons donc respectivement consacré les chapitres deux (cf. p. 13), trois (cf. p. 35) et quatre (cf. p. 87) à une revue de la littérature en matières de support informatisé à la coopération, de visualisation d'information et de visualisation appliquée à des situations de nature coopérative.

Dans le cinquième chapitre (cf. p. 125), nous proposons une nouvelle taxonomie des techniques de visualisation d'informations relatives à un contexte de coopération. Cette classification repose naturellement sur des travaux antérieurs. Elle est néanmoins originale car elle s'applique à un domaine bien particulier pour lequel nous n'avons pas connaissance de l'existence de travaux similaires.

Nous étudions ensuite plus en détail la nature des informations à visualiser dans un contexte de coopération (chapitre six, cf. p. 149). De nouveaux concepts théoriques sont proposés concernant la conscience de groupe. A la fin du chapitre, nous décrivons trois techniques de visualisation spécifiquement destinées à représenter certains aspects de situations de coopération supportées par des outils informatiques. La première représente l'évolution temporelle d'indices d'activité coopérative sous la forme d'un diagramme classique à deux dimensions. La seconde visualise la valeur courante de ces indices sous la forme d'un drapeau de couleur et de dimensions sémantiquement représentatives. Enfin, la troisième permet de visualiser les actions impliquant un acteur au cours d'une journée en utilisant la métaphore d'une horloge.

Le septième chapitre (cf. p. 179) est dédié à une technique de visualisation plus générique : l'ellimap. En premier lieu, les motivations qui ont conduit à son élaboration y sont décrits en détail. Ensuite, comme une partie significative de nos travaux fut allouée à l'amélioration successive de notre proposition initiale en la matière, nous avons estimé utile de relater ce cheminement. Nous mettons également en évidence les points forts et les lacunes des différentes approches que nous avons développées.

Compte tenu de son importance potentielle dans un grand nombre de contextes, nous avons ensuite voulu comparer cette nouvelle technique (ellimap) avec un de ses concurrents les plus sérieux : une variante de la technique des '*treemaps*' appelée '*squarified treemaps*'. Le huitième chapitre (cf. p. 237) relate notre démarche méthodologique, les évaluations que nous avons mises en œuvre et les résultats obtenus. Il décrit également l'environnement de test logiciel FESTIV qui fut développé spécifiquement pour notre évaluation.

Le neuvième chapitre (cf. p. 291) décrit les principales fonctionnalités du prototype logiciel qui implémente concrètement les concepts théoriques sous-jacents aux ellimaps. Nous y présentons également quelques exemples d'application de ce prototype pour visualiser des informations issues de plates-formes électroniques de support à la coopération.

Cette thèse se termine avec le dixième chapitre (cf. p. 311) qui résume les principales contributions de nos travaux et ouvrent des perspectives pour les poursuivre dans le futur.

Chapitre 2 : Le support informatisé à la coopération

« *Le plus grand plaisir humain est sans doute dans un travail difficile et libre fait en coopération, comme les jeux le font assez voir.* »

Alain

1. Les théories de l'activité coopérative

Dans cette section, nous présentons succinctement quelques théories parmi les plus importantes pour analyser des situations dans lesquelles se produisent des activités coopératives. Leur impact sur la conception de systèmes informatisés d'aide à la coopération est également discuté. Nous terminons cette présentation par une discussion et l'identification d'éléments clés que nous retenons de ces théories pour notre réflexion.

1.1 La théorie de l'activité

La théorie de l'activité est issue de l'école russe de la psychologie, en particulier des travaux précurseurs de Vygotsky (1978) et Leontiev (1978). Elle s'est ensuite répandue dans une communauté plus large.

Lonchamps (2003, pp. 44-49) explique que la théorie de l'activité se fonde sur un ensemble de principes de base qui constituent un cadre conceptuel :

- La structuration hiérarchique des activités en trois niveaux d'analyse :
 - les *activités* qui naissent d'une motivation concrétisée en un *objet* ;
 - les *actions* qui sont les composants des activités associés à des *buts* précis, elles sont réalisées consciemment ;
 - les *opérations* qui sont les moyens d'exécution des actions, elles sont des comportements routiniers.

L'activité est orientée par un *objet* qui est transformé en un *résultat* ; elle est réalisée par un *sujet* (individuel ou collectif) ; le rapport entre le sujet et l'objet est concrétisé par l'*outil*.

- Le caractère objectif de la réalité :
les éléments de la réalité possèdent des propriétés qui peuvent être étudiées avec des méthodes objectives.
- L'internalisation / externalisation des activités :
Il faut distinguer les *activités internes* (processus mentaux) et les *activités externes* en interaction avec la réalité. L'*internalisation* transforme des activités externes en activités internes alors que l'*externalisation* est le mécanisme inverse.
- Le rôle de médiation des *outils* entre le *sujet* et l'*objet* :
Les outils constituent une expérience accumulée pour résoudre un certain type de problème. L'expérience réside à la fois dans la structure de l'outil et dans la connaissance de son utilisation.
- La dynamique du développement des activités :
Les pratiques humaines sont des processus évolutifs où les niveaux individuel et social sont en interférence permanente. L'étude des activités doit donc se baser de préférence sur la participation active et l'observation des changements et du développement des acteurs.

1.2 La cognition distribuée

La théorie de la cognition distribuée a été développée par Edwin Hutchins et ses collègues (Hutchins, 1995 ; Hutchins et Klausen, 1996 ; Hollan et al., 2000). Elle s'intéresse aux systèmes socio-techniques, décrits comme un ensemble de personnes et d'*artefacts*, tous deux conceptuellement considérés comme des agents du système. Elle analyse les interactions entre ces composants dans le cadre où ils sont situés. Les questions essentielles qu'elle traite concernent la structure, le traitement et la circulation des représentations internes (présentes dans le cerveau des personnes) et externes (matérialisées par des *artefacts*, tels que plans, notes, calendriers, compte-rendu de réunions...).

En fait, elle étend le modèle de l'individu utilisé en psychologie cognitive pour considérer la cognition comme un phénomène intégré dans un cadre technologique et social. Dès lors, le cadre d'observation se déplace de phénomènes internes au cerveau humain vers des phénomènes se produisant dans un système socio-technique. L'observation directe de certaines représentations devient donc possible, comme, par exemple, la circulation de documents (représentation externe).

Cette approche permet d'étudier des phénomènes cognitifs qui ne sont pas observables si on se limite au seul niveau de l'individu. C'est notamment le cas du travail coopératif. Cette théorie a été essentiellement utilisée pour comprendre des situations de travail complexes mettant en œuvre des artefacts technologiques comme le pilotage d'avion (Hutchins et Klausen, 1996) ou l'ingénierie (Perry, 1997).

Au point de vue méthodologique, la cognition distribuée repose sur l'analyse :

- des contributions de l'environnement dans lequel les activités ont lieu,
- des médias associés aux représentations (p.ex. manuels, instruments, moyens d'affichage...),
- des interactions entre les individus,
- de l'utilisation interactive des *artefacts* par les individus.

DeCortis et al. (1997) proposent une présentation générale de cette théorie en isolant quatre concepts de base : l'horizon d'observation, le partage de la connaissance, les trajectoires de l'information et la compréhension intersubjective.

Le premier concept est celui d'*horizon d'observation*. DeCortis le définit comme l'espace de travail fonctionnel que chaque participant appréhende en plus de sa propre tâche ou comme la partie de la tâche qui peut être vue ou entendue par chaque membre de l'équipe. Hutchins et Seifert (1992) ont discuté les raisons pour lesquelles la taille de cet horizon d'observation était particulièrement importante.

- Elle a un impact sur la correction d'erreurs, en permettant ou limitant la surveillance par d'autres membres de l'équipe.
- Elle influence le processus d'apprentissage, en définissant la partie de l'environnement qui est disponible comme contexte d'apprentissage pour chacun des membres de l'équipe.
- Elle a un impact sur l'aide entre membres de l'équipe, en rendant visible le besoin d'aide par un participant donné.

DeCortis conclut sa présentation de ce concept en précisant que '*l'étendue avec laquelle un système peut bénéficier de la redondance fonctionnelle de ses parties dépend de l'étendue avec laquelle les parties sont conscientes du besoin d'un fonctionnement redondant*'. Cette notion est appelée '*redondance du suivi des activités*' par Lonchamps (2003, p. 62).

Le fait que la *connaissance* soit *partagée* constitue le second élément de base de la théorie de la cognition distribuée. Cette '*redondance des connaissances*' signifie que la connaissance nécessaire pour accomplir la tâche n'est pas distribuée de manière exhaustive et mutuellement exclusive entre les individus. Chaque personne doit partager sa connaissance en coopérant avec les autres dans le but d'accomplir la tâche coopérative. De plus, le recouvrement partiel

entre les connaissances des individus augmente la fiabilité du système socio-technique de coopération.

La troisième notion fait référence aux *chemins et trajectoires de l'information* dans le système socio-technique. Dans un tel système, la trajectoire des informations est imprévisible. En effet, l'accès à l'information et son traitement sont distribués parmi les membres du groupe.

Au cours de ces parcours divers, l'information change de support (p.ex. voix humaine, courriel, document sur papier) et de nature (représentation interne / externe). Zhang et Norman (1994) ont étudié en particulier les représentations externes et ont identifié différents rôles qu'elles peuvent jouer :

- l'aide à la mémorisation, en étant affichées en face des acteurs (p.ex. tableaux de bord, indicateurs...) ;
- la mise à disposition d'information sous une forme directement perceptible et utilisable ;
- l'incitation au mode d'utilisation de l'information (*affordance*) ;
- l'ancrage et la structuration du comportement cognitif en établissant des contraintes qui permettent ou interdisent certains comportements.

Pour terminer sur ce point, on peut conclure que les représentations externes peuvent rendre une tâche plus facile mais qu'elles peuvent aussi changer la nature de cette tâche.

La *compréhension intersubjective* est le quatrième concept qu'il est important de présenter. Une équipe constitue une communauté de pratique. Au sein de celle-ci, les membres partagent en général une connaissance sur la manière dont les choses sont sensées se dérouler et sur leur déroulement habituel. Au cours de leurs interactions, ils utilisent cette connaissance partagée pour construire une compréhension commune d'une situation particulière, appelée compréhension intersubjective. Grâce à celle-ci, des modes très efficaces de communication sont mis en œuvre sous la forme, par exemple, de comportements non verbaux ou d'interprétation des éléments implicites d'énoncés.

Plusieurs concepts de la *théorie de la cognition distribuée* peuvent être utilisés dans le cadre de la conception d'outils informatiques collaboratifs.

Le concept d'horizon d'observation peut aider à délimiter la visibilité des objets ou des interactions sur les objets disponibles sur une plate-forme électronique de coopération. La théorie de la cognition distribuée attire en effet l'attention sur l'utilité de mettre à disposition des individus des informations qui ne sont pas directement reliés à leurs propres tâches.

A un autre niveau, la prise en compte de la compréhension intersubjective des intervenants peut influencer la nature des opérations qu'il est souhaitable d'automatiser par un logiciel ainsi que l'extension de cette automatisation. De ce point de vue, la démarche de conception d'un système informatique collaboratif devrait intégrer une réflexion sur la manière de conserver des modes implicites de communication, ce qui peut se traduire, dans certains cas, en offrant un équivalent électronique.

Par ailleurs, le rôle décisif joué par les représentations externes peut également être intégré dans la démarche de conception de logiciels collaboratifs. Certains éléments de ceux-ci et en particulier les interfaces homme-machine constituent indéniablement de telles représentations externes qui devraient être capables de remplir les rôles identifiés par Zhang et Norman. Un logiciel collaboratif pourrait également offrir une fonctionnalité de transformation des représentations externes, qui pourraient prendre la forme, par exemple, de diverses techniques de visualisation de l'information.

1.3 La théorie de la coordination

La théorie de la coordination (Malone et Crowston, 1990 ; Malone et Crowston, 1994 ; Rognin, 1997 ; Malone et al., 1999) fut développée au début des années 1990 par Thomas Malone et son équipe du *Centre for Coordination Science*, au *Massachusetts Institute of Technology*.

Ses fondateurs, Malone et Crowston (1990) la définissent comme *‘un ensemble de principes expliquant comment des activités peuvent être coordonnées, c.-à-d. comment des acteurs peuvent travailler ensemble harmonieusement²’*.

Cette approche interdisciplinaire utilise et combine des méthodes et résultats issus de la théorie de l’organisation, de l’économie, de l’informatique et de la psychologie. Elle a pour objet l’étude de la coordination au sein de systèmes variés.

La théorie de la coordination étudie les processus organisationnels en considérant que ceux-ci dépendent des mécanismes de coordination choisis pour gérer les dépendances entre les tâches et les ressources impliquées dans le projet. Ces mécanismes consistent principalement en des traitements d’information. Par conséquent, la question sous-jacente est donc d’étudier comment la large diffusion des technologies de l’information change la manière avec laquelle les personnes travaillent ensemble.

Dans cette théorie, la coordination est analysée en termes d’*acteurs* qui réalisent des activités interdépendantes pour atteindre des *buts* donnés. Ces activités peuvent nécessiter l’utilisation de *ressources*. Le concept de dépendance s’avère donc central. Parmi plusieurs propositions de classification des différents types de dépendances, Malone et ses collègues (1999) ont notamment proposé une taxonomie basée sur trois catégories fondamentales :

- Les dépendances de flux, qui existent quand une activité produit une ressource qui doit être utilisée par une autre activité.
- Les dépendances de partage, correspondant à l’utilisation d’une même ressource par plusieurs activités différentes.
- Les dépendances d’ajustement, qui existent dès lors que des activités produisent une même ressource en associant leurs compétences.

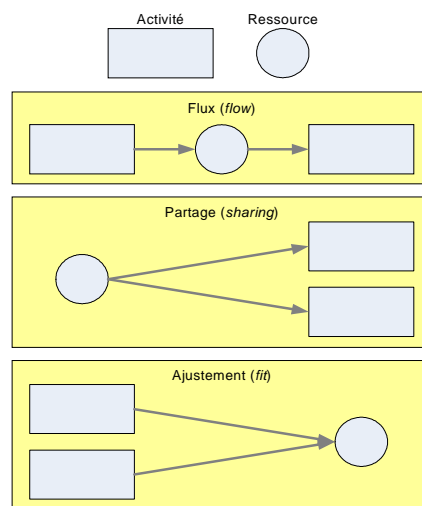


Fig. 3: Types de dépendances selon Malone et al.

Selon cette théorie, plusieurs mécanismes de coordination peuvent être associés à un type de dépendance donné. Les différents choix opérés conduiront à des formes organisationnelles

² *‘...a body of principles about how activities can be coordinated, that is, about how actors can work together harmoniously’*

différentes ou à des processus différents. La théorie de la coordination prône donc la revue systématique des différentes options en vue de remplacer les mécanismes moins performants par d'autres plus efficaces (cf. réingénierie des processus). Malone et ses collègues ne réduisent cependant pas la problématique de la coordination à la seule gestion des dépendances. Ils mentionnent également l'importance des prises de décision et des modes de communication.

Selon ses auteurs, la théorie de la coordination possède deux objectifs principaux : la compréhension de l'influence des technologies de l'information sur les organisations et les marchés d'une part, et l'aide à la conception d'outils permettant le travail coopératif. Compte tenu de notre champ d'étude, sommes principalement concernés par le second objectif.

Dans ce cadre de notre réflexion sur les outils informatique utilisés dans un contexte coopératif, la *théorie de la coordination* présente un intérêt certain. En effet, elle permet de classer les systèmes selon les mécanismes de coordination qu'ils mettent en œuvre. Elle peut aussi susciter le développement et l'utilisation de nouveaux systèmes, en établissant des analogies avec des mécanismes de coordination issus d'autres disciplines que celle qui est étudiée.

1.4 L'ergonomie cognitive

L'ergonomie cognitive est issue de travaux initiés par l'école des ergonomes français tels que Montmollin, Leplat, Savoyant... (cf. Decortis et al., 1997 pour une présentation générale de cette approche). Elle s'intéresse à l'étude de situations de travail où des individus effectuent une certaine tâche en s'aidant de différents *artefacts*. Elle se focalise sur les activités cognitives des individus et leurs environnements de travail cognitif (cf. Lonchamps, 2003, p. 85). Il s'agit d'une approche essentiellement pragmatique, reposant sur des observations de terrain.

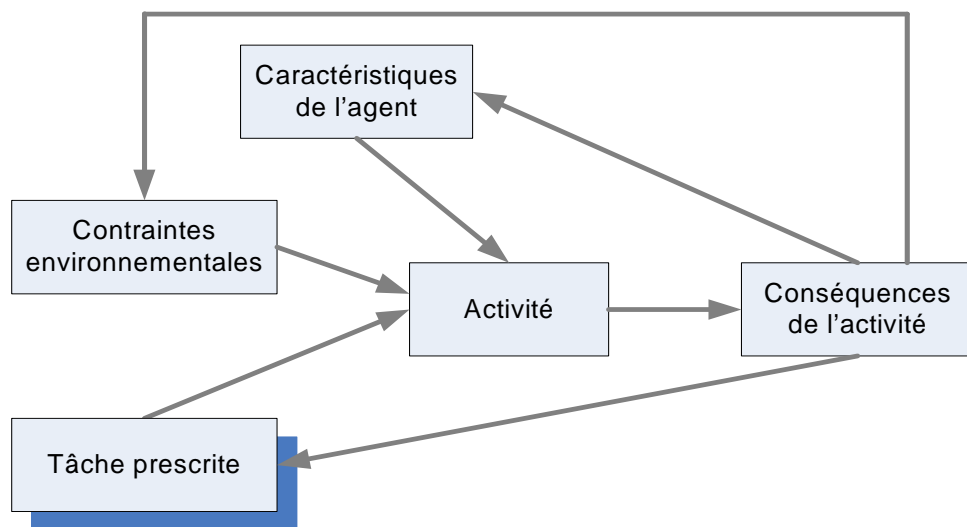


Fig. 4: Relation entre tâche prescrite et activité réelle (source : [123], p. 87)

Un élément spécifique nous semble particulièrement intéressant dans cette approche : la différenciation entre *tâche prescrite* et *activité effective*. La tâche désigne généralement ce qui est à faire par l'opérateur et comment il doit le faire. Elle se caractérise par un but donné et des conditions d'obtention déterminées par l'organisation. L'activité est la réponse individuelle à la tâche prescrite. Elle dépend des compétences de l'individu, de son intelligence, de ses capacités d'adaptation et d'autonomie, de ses stratégies cognitives, de ses mécanismes d'adaptation et d'anticipation, etc. (cf. Fig. 4) 'Cette distinction tâche/activité permet de souligner le caractère non totalement prescriptible et informel de beaucoup d'activités de travail. En effet, face aux exigences constantes de la tâche, des variations

apparaissent dans la mise en œuvre effective, liées soit aux variations interindividuelles soit aux variations de la situation de travail. (Lonchamps, 2003, p. 88).

Cette distinction ‘tâche – activité’ apparaît non seulement dans les activités individuelles mais également dans les activités collectives.

Un autre élément développé par l’ergonomie cognitive présente un réel intérêt dans le cadre de notre réflexion : l’importance accordée à la notion de représentation commune et à son rôle dans la coordination des activités.

Ce concept que Decortis et ses collègues (1997) appellent ‘*représentation distribuée*’ fut présenté avec de subtiles variantes par d’autres auteurs sous d’autres termes tels que ‘*référentiel commun*’ (de Terssac et Chabaud, 1991), ‘*représentations fonctionnelles communes*’ (Leplat, 1991), ‘*environnement cognitif mutuel*’ (Sperber et Wilson, 1987), ‘*environnement mutuel partagé*’ (Pavard et al., 1994).

Les *représentations distribuées* sont définies comme des ‘*représentations fonctionnelles qui guident et contrôlent une activité réalisée collectivement*’ (Leplat, 1991). Selon Decortis et al. (1997), elles constituent en fait ‘*l’intersection entre les représentations fonctionnelles de plusieurs opérateurs contribuant à une tâche*’ et ‘*elles devraient contenir un aspect figuratif représenté par les états du processus et un aspect opératoire caractérisé par les opérations qui doivent être réalisées*’.

Les représentations distribuées ne sont pas statiques mais elles évoluent avec l’expérience. Elles sont élaborées à partir des tâches prescrites et se modifient en fonction des activités réelles. Elles se construisent progressivement à travers les échanges entre les acteurs.

2. Le travail coopératif assisté par ordinateur

2.1 Définition

Le terme de Travail Coopératif Assisté par Ordinateur (TCAO ou CSCW en anglais pour *Computer Supported Cooperative Work*) fut proposé en 1984 par Irène Greif et Paul Cashman lors d’un séminaire au Massachusetts Institute of Technology (cf. Grudin, 1994). La première conférence dédiée au CSCW fut organisée en 1986, à Austin (Texas).

Différentes définitions ont été proposées pour délimiter la portée de cette discipline.

1. Le CSCW a aussi été considéré comme une ‘*tentative pour comprendre la nature et les caractéristiques du travail coopératif avec comme objectif la conception d’une technologie informatique adéquate*’ (Bannon et Schmidt, 1991).
2. Ellis, Gibbs et Rein (1991) expliquent que ‘*le CSCW regarde comment les groupes travaillent et cherche à découvrir comment la technologie (en particulier les ordinateurs) peut les aider à travailler*’.
3. Decortis et al. (1997) mentionnent que ‘*[la discipline du CSCW] couvre tout ce qui est fait avec le support d’un ordinateur dans le cadre d’activités dans lesquelles plus d’une personne est impliquée*’.
4. Olson et Olson (2003) définissent la discipline du CSCW comme ‘*l’étude de comment les personnes travaillent ensemble en utilisant les technologies de l’information et de la communication*’.
5. Longchamp (2003, p.15) utilise le terme générique de système coopératif pour qualifier ‘*tout système informatisé visant à assister un groupe d’utilisateurs qui travaillent ensemble et interagissent dans le but de réaliser une tâche commune*’.

Le terme ‘*Groupware*’ est souvent utilisé pour nommer des concepts proches du CSCW ou qui y sont, pour le moins, particulièrement reliés.

6. Ellis, Gibbs et Rein (1991) précisent que ‘*le but du groupware est d’assister les groupes de personnes dans leur communication, leur collaboration et la coordination de leurs activités*’. Ils définissent le groupware comme ‘*les systèmes basés sur*

ordinateur, qui supportent les groupes de personnes engagés dans une tâche (ou un but) commune et qui fournissent une interface pour un environnement partagé’.

7. George (2003) définit le concept de *groupware* comme ‘une classe de logiciels qui aident les membres d’un groupe à mieux travailler ensemble’.

2.2 Modèle des 3C (Ellis)

Le ‘modèle des 3C’ est un des modèles de référence pour conceptualiser le Travail Coopératif Assisté par Ordinateur. Il fut proposé par Ellis et ses collègues (Ellis et al. 1991 ; Ellis, 1994 ; George, 2003). Selon ce modèle, le travail de groupe peut être caractérisé par le type de relations existant entre les acteurs. Trois étapes, correspondant à des degrés d’intensité croissants, ont été identifiées.

En premier lieu, un travail de groupe nécessite un certain niveau de **communication**, c.à.d. de partage d’information.

Ensuite, un travail de groupe demande une certaine **collaboration** entre les membres du groupe. Naturellement, la collaboration exige un niveau suffisant de communication.

Enfin, l’efficacité de la communication et de la collaboration sera améliorée si les activités du groupe font l’objet d’une **coordination**.

Sur cette base, on peut différencier les outils rencontrés dans les logiciels de *groupware* selon qu’ils sont dédiés à remplir l’une ou l’autre de ses fonctions.

1. Les outils de communication supportent le partage d’information entre les membres du groupe. Ils permettent à chaque membre de donner accès à certaines informations.
Exemple : Agenda partagé
2. Les outils de collaboration se focalisent sur l’utilisation de l’information partagée pour créer des solutions aux problèmes que le groupe doit résoudre.
Exemple : Systèmes d’aide à la décision de groupe
3. Les outils de coordination se concentrent sur la gestion du travail que le groupe est en train d’accomplir.
Exemple : Système de Workflow

2.3 Typologie des outils de support au travail coopératif

Un grand nombre de logiciels, très diversifiés, peuvent être catégorisés comme appartenant au monde du TCAO (CSCW). On peut citer, à titre d’exemples, la messagerie électronique, les systèmes de partage de documents, la vidéoconférence, la messagerie instantanée, les systèmes de support aux réunions... Très rapidement, est apparue la nécessité de les classifier pour faire émerger des caractéristiques génériques communes.

L’approche spatio-temporelle, que Johansen (1988) a résumée, dès 1988, dans sa célèbre représentation matricielle (cf. Fig. 5) fait figure de référence incontournable à ce niveau. Johansen distingue ainsi quatre cas selon que les membres du groupe interagissent en étant dans le même lieu ou dans des lieux différents (axe spatial) et selon que ces interactions soient synchrones ou asynchrones (axe temporel).

Temps	Espace	
	Identique	Différent
Différent	<i>Support for shift work ...</i>	<i>E-mail...</i>
Identique	<i>Meeting support...</i>	<i>Video conferencing...</i>

Fig. 5: la matrice de Johansen

Quelques années plus tard, Johnson et Johnson (2001) ont développé l’idée à la base de la matrice de Johansen pour affiner la classification des différents types d’activités collaboratives. Ils identifient trois axes de classification : les parties en présence, la prise de

conscience de la collaboration et l'aspect spatio-temporel des interactions se produisant au cours de la collaboration.

En ce qui concerne les parties en présence, ces auteurs identifient trois types d'acteurs : les personnes, les agents logiciels et les organisations.

Table 1: Classification des collaborations selon les parties en présence

Acronyme	Parties en présence	Remarque
PPC (<i>Person-Person Collab.</i>)	Personne – Personne	Peut se produire entre personnes ayant des expertises, des responsabilités différentes.
PAC (<i>Person-Agent Collab.</i>)	Personne – Agent	L'agent peut soit réduire la charge de travail associée à la tâche concernée, soit il peut ajouter de la connaissance pour améliorer l'accomplissement de la tâche.
POC (<i>Person-Org. Collab.</i>)	Personne – Organisation	La collaboration ne nécessite pas que les parties soient identifiées en tant qu'individus.
AAC (<i>Agent-Agent Collab.</i>)	Agent – Agent	-
AOC (<i>Agent-Org. Collab.</i>)	Agent – Organisation	-
OOC (<i>Org.-Org. Collab.</i>)	Organisation – Organisation	La collaboration est souvent placée dans le cadre d'une relation formellement définie.

Les activités de collaborations peuvent aussi être distinguées selon la prise de conscience qu'ont les acteurs de cette collaboration.

Table 2: Classification des collaborations selon la conscience qu'en ont les acteurs

Acronyme	Conscience de la collaboration
MGS (<i>Main Goal Sharing</i>)	Les participants sont conscients de la collaboration et travaillent consciemment ensemble vers un but commun.
DCS (<i>Distributed Contributing Goals</i>)	Des participants accomplissent ensemble des tâches dont ils ignorent qu'elles s'inscrivent dans une démarche ayant un but commun et dont ils ignorent comment elles participent à cet objectif plus large.
IGS (<i>Independent Contributing Goals</i>)	Des participants réalisent individuellement et de leur propre initiative des tâches qui ne sont pas des activités de collaboration en tant que telles mais qui contribuent à atteindre un but commun dans le cadre d'une collaboration.

Enfin, Johnson et Johnson adaptent la matrice de Johansen pour caractériser la collaboration au niveau spatial et temporel. Ils remplacent la distinction entre les activités synchrones et asynchrones par la distinction entre activités instantanées et différées.

Table 3: Classification des collaborations selon les aspects spatio-temporels

Acronyme	Aspects spatio-temporels
IPI (<i>Immediate – Proximal Interaction</i>)	Les parties concernées sont situées au même endroit et les interactions entre elles sont instantanées.
IDI (<i>Immediate – Distant Interaction</i>)	Les parties concernées sont situées dans des endroits différents et les interactions entre elles sont instantanées.
DDI (<i>Delayed – Distant Interaction</i>)	Les parties concernées sont situées dans des endroits différents et les interactions entre elles sont différées.
DPI (<i>Delayed – Proximal Interaction</i>)	Les parties concernées sont situées au même endroit et les interactions entre elles sont différées.

Ces éléments permettent de structurer la réflexion quand on vise à concevoir de nouveaux outils informatiques d'aide au travail coopératif. En effet, il est important de bien préciser dans quel(s) type(s) de travail coopératif l'outil est destiné à être utilisé.

3. Le concept de conscience de groupe

3.1 Définitions

Le concept de conscience de groupe (traduction du terme anglais : '*awareness*') constitue une composante essentielle de l'étude du travail coopératif. En 1994, Ellis (1994) affirmait : '*la philosophie du groupware est d'encourager la coopération en la rendant connue et instantanément apparente à tous ceux qui partagent quelque chose avec quelqu'un*'.

En fait, dès 1992, Paul Dourish et Victoria Bellotti (1992) figuraient parmi les premiers à introduire ce concept. Ils définissaient la conscience de groupe comme '*une compréhension de l'activité des autres, qui fournit un contexte pour sa propre activité. Ce contexte est utilisé pour s'assurer que les contributions individuelles sont pertinentes pour l'activité du groupe dans son ensemble et pour évaluer les actions individuelles par rapport aux objectifs et à l'évolution du groupe. L'information permet alors aux groupes de gérer le processus de travail collaboratif*'.

Endsley (1995, p. 36) définit le concept de conscience de groupe (*awareness*) comme le fait de '*savoir ce qui se passe*'.

On pourrait croire que ce concept est unique et bien délimité dans la littérature. Cependant, Greenberg et ses collègues (1996) observent que le concept de conscience de groupe fait référence à plusieurs notions différentes :

- Connaissances informelles liées à la communauté de travail (*informal awareness*)
 - Qui est là ?
 - Que pense-t-il faire ?
- Informations de la sphère sociale qu'une personne maintient à propos des autres (*social awareness*)
 - Est-ce que Mr X est attentif ?
 - Quel est son état émotionnel ?
 - Quel est son niveau d'intérêt ?

- Informations sur la composition, l'histoire et le fonctionnement du groupe (*group-structural awareness*)
 - Quel statut Mr X possède-t-il ?
 - Quelle est sa position sur cette question ?
 - Quels sont les processus internes du groupe ?
- Connaissances qu'une personne a du travail commun au sein de l'espace partagé et de l'état des artefacts partagés (*workspace awareness*)
 - Qui est présent dans l'espace de travail partagé ?
 - Où sont localisées les personnes présentes dans l'espace de travail partagé ?
 - Que font les personnes présentes dans l'espace de travail partagé ?

Par ailleurs, différents travaux (cf. Norman, 1993 ; Adams et al., 1995 ; Endsley, 1995) ont mis en évidence quatre caractéristiques de base du concept de conscience de groupe :

- la conscience de groupe est la connaissance de l'état d'un environnement délimité dans l'espace et le temps ;
- comme l'environnement change au cours du temps, la conscience de groupe est une connaissance qui doit être maintenue à jour ;
- la conscience de groupe est entretenue par l'interaction des personnes avec l'environnement ;
- la conscience de groupe est un but secondaire associé à une tâche à accomplir dans l'environnement.

3.2 Paradigmes de mise en œuvre

Lonchamps (2003, p. 206-208) met en évidence le fait que deux paradigmes sont le plus souvent utilisés pour implémenter en pratique le concept de conscience de groupe : le modèle « Publication / Souscription » et le modèle spatial. Ces deux approches sont décrites dans les paragraphes suivants.

3.2.1. Modèle Publication / Souscription

La Fig. 6 illustre le modèle « Publication / Souscription ». Les changements d'états de certains objets ou de leur environnement sont matérialisés par des événements (messages). Les événements dont la survenance est publique sont communiqués à un intermédiaire (étape de publication) qui les transmet à d'autres objets qui ont préalablement manifesté leur intérêt pour ce type d'événements (étape de souscription).

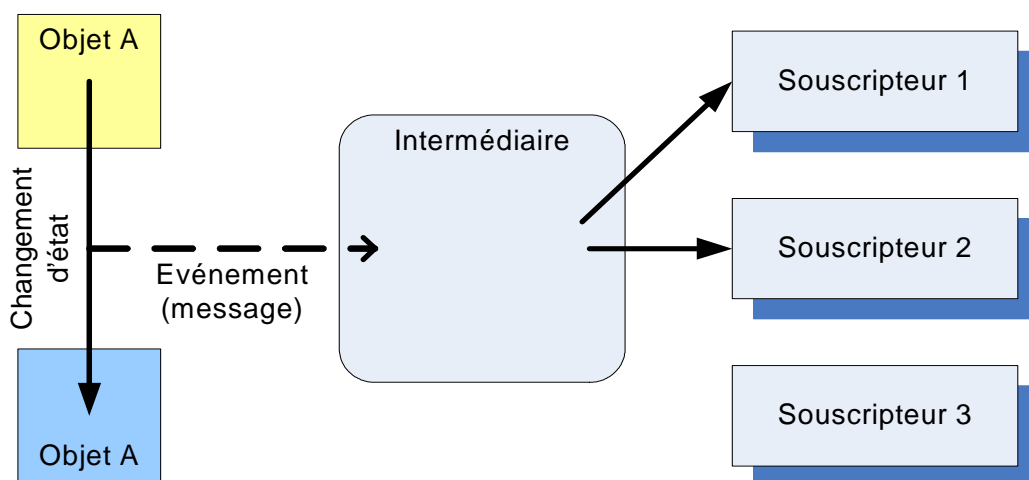


Fig. 6: Modèle Publication / Souscription

La souscription peut se baser sur deux types d'information : le sujet du message (c.-à-d. le type d'événement) ou le contenu du message. Dans le premier cas, les événements sont associés, lors de leur production, à (au moins) un thème et la souscription consiste à choisir les thèmes qui présentent un intérêt pour le souscripteur. Dans le second cas, les souscripteurs expriment leurs intérêts sous la forme de requêtes sur le contenu des messages.

3.2.2. Modèle spatial

Ce modèle a été décrit de manière théorique par Rodden (1996). Il constitue une généralisation de travaux antérieurs de Benford et ses collègues (1995). Le modèle « Focus et Nimbus » repose sur des notions associées au positionnement spatial, telles que la proximité ou la distance.

Ce modèle offre un moyen de déterminer le niveau de conscience de groupe qui devrait être présent pour deux acteurs dans un espace partagé. Il repose sur l'hypothèse d'une relation inverse entre distance et conscience de groupe. Au plus une personne est loin des autres, au moins elle est consciente de leur présence et de leur activité.

Pour Benford et ses collègues, les objets dans l'espace sont responsables pour contrôler leurs interactions sur base de niveaux quantifiables de présence de groupe entre eux. La conscience de groupe entre les objets est manipulée via deux sous-espaces, *focus* et *nimbus*, dans lesquels les objets choisissent de diriger respectivement leur présence ou leur attention.

- Au plus un objet A est dans le *focus* d'un objet B, au plus B est conscient de A.
- Au plus un objet A est dans le *nimbus* d'un objet B, au plus A est conscient de B.

Ces chercheurs proposent alors que le niveau de conscience de groupe qu'un objet X a d'un objet Y dans un milieu M est une fonction du *focus* de X dans M et du *nimbus* de Y dans M.

Rodden (1996) a étendu ce modèle. Il considère un espace partagé (*shared space*) habité par un ensemble d'objets O et un ensemble d'utilisateurs U. La position d'un utilisateur dans l'espace est décrite par un objet (*position objet*) et par un ensemble d'objets adjacents (*adjacency set*). La fonction *focus* associe à un utilisateur différentes positions qui correspondent à ses différents « centres d'intérêt ». La fonction *nimbus* associe de la même manière différentes positions qui correspondent aux objets « intéressés par cet utilisateur ». Un même utilisateur peut avoir plusieurs *foci* et *nimbi*, associés par exemple à différents médias. Le niveau de conscience de groupe entre deux utilisateurs est fonction du recouvrement de leurs *focus* et *nimbus* agrégés respectifs. Pour chaque application, ce niveau de conscience de groupe doit être calculé selon une formule spécifique à concevoir.

Une autre évolution de ces travaux fut proposée par Baudel (1995). Il définit les concepts de base de son modèle comme suit :

- Le *nimbus* est l'environnement perçu par un agent à un instant donné.
- L'*aura* d'un agent est l'ensemble des manifestations de l'agent permettant d'en déduire une partie de son état interne.
- L'agent est aussi caractérisé par son *focus* ou point d'attention de l'agent lors de la production intentionnelle d'un changement d'état de l'environnement perçu. Le *focus* se situe dans l'intersection de l'*aura* et du *nimbus* de l'agent : il est nécessaire d'avoir une conscience de ses capacités d'action pour agir intentionnellement sur un autre objet ou un agent.

Baudel explique que « *un système interactif se doit de maximiser son aura et son espace perçu afin que l'utilisateur n'ait qu'à fournir un effort minimal d'accommodation de ses propres ressources. En d'autres termes, la bande passante humain – machine se doit d'être potentiellement restreinte mais bien exploitée, tandis que le bande passante machine – humain se doit d'être aussi large que possible, afin de mettre le maximum d'information à disposition de l'utilisateur.* »

Les moyens informatiques pour supporter la conscience de groupe doivent donc tenir compte des capacités cognitives limitées de l'être humain.

3.2.3. Modèle de traitement de la connaissance de Brézillon et al.

Brézillon, Borges, Pino et Pomerol (2004) ont proposé un modèle de traitement de la connaissance contextuelle dans le travail de groupe (cf. Fig. 7). Les personnes créent de la connaissance au niveau individuel. Celle-ci est communiquée au reste du groupe, présentée via une interface utilisateur et éventuellement stockée.

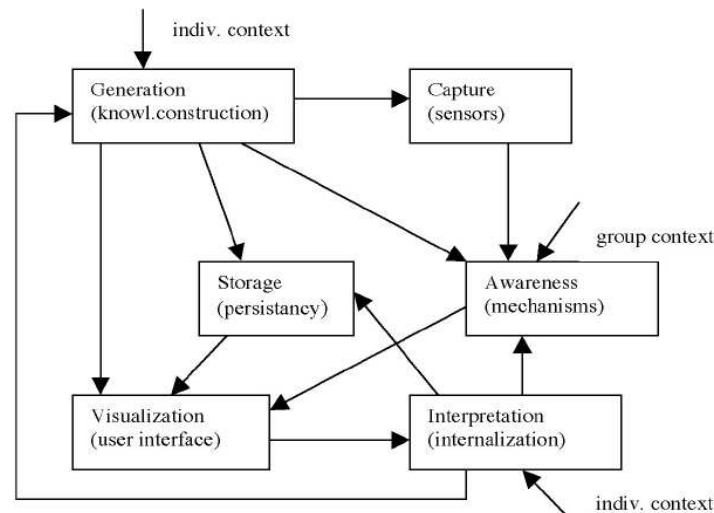


Fig. 7: Modèle de Brézillon et al. (source : [30])

L'étape de *génération* (*generation step*) consiste en la fourniture d'information au groupe par une personne. Cette information peut être du contenu pour les résultats du groupe (*group output*) ou de l'information associée, comme des questions ou des suggestions. Une partie de cette information est stockée, selon des conditions pré-établies.

L'étape de support à la conscience de groupe (*awareness step*) est le moyen par lequel l'information est fournie aux autres participants. Il y a plusieurs entrées (*inputs*) dans ce processus :

- de l'information issue de l'étape de génération,
Cette information peut nécessiter d'être transformée (p.ex. résumée, filtrée) avant d'être rendue publique aux autres membres du groupe. Ces transformations tiennent compte de spécifications sur le traitement données par les utilisateurs individuels.
- de l'information issue de l'étape de capture (*capture step*).
Cette information doit probablement faire l'objet d'un traitement pour éviter une surcharge informationnelle.
- de l'information issue de l'étape de stockage (*storage step*)
- de l'information sur le contexte du groupe.

3.3 Workspace Awareness Framework

3.3.1. Contexte

Carl Gutwin et Saul Greenberg, du *Interaction Lab*, de l'Université de Saskatchewan au Canada, figurent parmi les spécialistes du concept de conscience de groupe (cf. p. ex. Gutwin et Greenberg, 1998 ; Gutwin et Greenberg, 1999 ; Gutwin et Greenberg, 2002 ; Gutwin et al., 2004). Dans un de leurs articles sur ce sujet (Gutwin et Greenberg, 2002), ils proposent un cadre général d'étude (*Workspace Awareness Framework*) de la conscience de groupe associée à un espace de travail partagé (*shared workspace awareness*). Ce résultat est le résultat de nombreuses années de travail sur cette thématique.

Leur but avoué est de fournir aux concepteurs de systèmes de *groupware* une théorie descriptive du concept de conscience de groupe pour les aider à imaginer de nouveaux systèmes. Gutwin et Greenberg se proposent d'aider ceux-ci à répondre aux trois questions fondamentales qui se posent lorsque l'on doit supporter la conscience de groupe avec un système de *groupware* :

- quelle information rassembler et distribuer ?
- comment présenter l'information au groupe ?
- quand l'information est-elle la plus utile ?

Gutwin et Greenberg précisent que leur théorie s'adresse à un type bien particulier de situations : de petits groupes qui travaillent dans un espace de travail partagé de taille moyenne. Plus précisément, ils délimitent le champ d'application de leur théorie avec quatre éléments :

- les types de groupes concernés
 - petits groupes de deux à cinq personnes
 - personnes engagées dans une collaboration à focalisation multiple (*mixed-focus collaboration*), qui passent fréquemment d'une activité individuelle à une activité partagée
- l'espace de travail où la collaboration se produit
 - surfaces planes de taille moyenne sur lesquelles des objets peuvent être placés et manipulés et autour desquelles un petit groupe de personnes peut collaborer
- les types de tâches que les groupes entreprennent
 - tâches de génération
 - tâches d'exécution
- les types de systèmes de *groupware* qui sont utilisés
 - systèmes distribués en temps réel qui permet aux personnes de travailler ensemble en même temps à partir de localisations différentes

Gutwin et Greenberg mentionnent quelques exemples de telles situations : la génération d'idées sur un tableau par un groupe de recherche ou la planification d'un calendrier de projet par des managers.

Ils mentionnent également que la théorie qu'ils proposent ne concerne qu'un type particulier de conscience de groupe, celle relative à un espace de travail (*workspace awareness*). Ils définissent cette notion comme '*la connaissance immédiate de l'interaction d'une autre personne avec un espace de travail partagé*³'. Elle concerne donc la connaissance de l'interaction des personnes avec l'espace de travail et pas seulement la connaissance de l'espace de travail seul. De plus, elle se limite aux événements qui se produisent dans cet espace de travail.

³ *up-to-the-moment understanding of another person's interaction with a shared workspace*

Selon eux, la conscience de l'espace de travail nécessite la connaissance de trois types d'information : où les autres travaillent, ce qu'ils font et ce qu'ils feront ensuite.

Gutwin et Greenberg positionnent leur théorie comme un complément à l'approche 'Focus/Nimbus'. Selon eux, celle-ci s'intéresse en priorité à des espaces contenant beaucoup de personnes alors que leur théorie vise les groupes plus petits.

3.3.2. Description

Le cadre d'analyse de Gutwin et Greenberg se divise en trois composants complémentaires :

- quelle information permet de construire une conscience de groupe de l'espace de travail ?
- comment cette information est-elle collectée ?
- comment la conscience de groupe de l'espace de travail est-elle utilisée pour la collaboration ?

Nous explicitons chacun de ces aspects dans les paragraphes suivants.

a. Caractérisation de l'information nécessaire

Après avoir analysé la littérature, Gutwin et Greenberg concluent qu'un ensemble d'informations de base sont nécessaires pour construire une conscience de groupe de l'espace de travail. Ces informations sont celles qui permettent de répondre aux questions : « qui, quoi, où, quand, comment ». Gutwin et Greenberg les ont regroupées en deux sous-ensembles, selon qu'elles soient relatives au présent (cf. Table 4) ou au passé (cf. Table 5).

Gutwin et Greenberg n'ont pas inclus dans leur théorie d'éléments associés au futur. Ils justifient ce choix en précisant qu'il est peu probable que les concepteurs de systèmes soient capables de supporter la maintenance de ces éléments, notamment parce que les informations sur le futur reposent sur des inférences, des extrapolations ou des prédictions.

Nous ne sommes pas entièrement d'accord avec cette position. Certains éléments du futur pourraient sans doute être utilement considérés dans leur cadre d'analyse. Par exemple, les événements très probables tels que des réunions ou des délais dans des projets pourraient s'avérer utiles pour compléter la conscience de groupe de l'espace de travail.

Table 4: Eléments de la conscience de groupe de l'espace de travail relatifs au présent (source : [83], p. 421)

Catégorie	Élément	Questions spécifiques
Qui ?	Présence	Y a-t-il quelqu'un dans l'espace de travail ?
	Identité	Qui participe ? Qui est-il ?
	Auteur	Qui fait cela ?
Quoi ?	Action	Que font-ils ?
	Intention	Quel est l'objectif de cette action ?
	Artefact	Sur quel objet travaillent-ils ?
Où ?	Localisation	Où travaillent-ils ?
	Regard (<i>gaze</i>)	Où regardent-ils ?
	Vue	Que peuvent-ils voir ?
	Portée (<i>reach</i>)	Où peuvent-ils modifier des choses ?

Table 5: Eléments de la conscience de groupe de l'espace de travail relatifs au passé (source : [83], p. 422)

Catégorie	Elément	Questions spécifiques
Comment ?	Historique d'actions	Comment cette opération s'est-elle produite ?
	Historique d'artefacts	Comment cet artefact est-il arrivé dans cet état ?
Quand ?	Historique des événements	Quand cet événement s'est-il produit ?
Qui ? (passé)	Historique des présences	Qui était là, et quand ?
Où ? (passé)	Historique des localisations	Où une personne est-elle allée ?
Quoi ? (passé)	Historique des actions	Qu'est ce qu'une personne a fait ?

Gutwin et Greenberg n'incitent pas à intégrer tous ces éléments de manière égale dans l'interface d'un système de *groupware*. Selon eux, deux éléments sont à prendre en compte pour déterminer quels éléments sont prioritaires. Premièrement, le degré d'interaction entre les participants à l'activité renseigne sur le niveau de spécificité de l'information (générale – spécifique) requis dans l'interface. Deuxièmement, la dynamique avec laquelle l'information change au cours du temps indique à quel rythme l'interface doit être mise à jour et renseigne sur la nécessité d'une présence explicite dans celle-ci.

b. Mode de collecte de l'information

Ce second volet de la théorie de Gutwin et Greenberg repose sur une observation de bon sens. Il faut essayer de rendre la maintenance des informations associées à la conscience de groupe aussi simple et directe que possible. Atteindre cet objectif sera facilité si les personnes peuvent rassembler l'information d'une manière qui leur est familière.

Gutwin et Greenberg indiquent qu'il existe trois sources principales d'information relatives à la conscience de groupe de l'espace de travail, auxquelles sont associés trois mécanismes de collecte de celle-ci.

La première source d'information est constituée par le corps des autres personnes dans l'espace de travail, c.-à-d. par exemple, leur posture ou le mouvement des bras, des mains ou des yeux. Le mécanisme de collecte associé est appelé communication résultante (*consequential communication*). Il a été défini par Segal (1994) comme '*le transfert d'information qui émerge en tant que conséquence de l'activité d'une personne dans un environnement*'. Ce type de communication corporelle est non-intentionnel mais il fournit une grande quantité d'information.

Les artefacts présents dans l'espace de travail représentent la seconde source d'information. Ils fournissent de l'information par le biais, par exemple, de leur apparence physique ou leur positionnement spatial. Le mécanisme de collecte associé est appelé '*feedthrough*'. Quand des artefacts sont manipulés ils envoient des informations et ce qui devrait normalement être communiqué comme feedback à la personne qui réalise cette action peut aussi informer d'autres personnes présentes (*feedthrough*).

La troisième source d'information consiste en la conversation et la gestuelle. Le mécanisme de collecte est appelé 'communication intentionnelle' (*intentional communication*). L'information peut être collectée à partir de conversations verbales de trois manières. Tout d'abord, le sujet même de la conversation peut être la conscience de groupe (p.ex. en demandant à quelqu'un : « que fais-tu ? »). En second lieu, on peut collecter de l'information sur la conscience de groupe en entendant par hasard une conversation d'autres personnes.

Enfin, on peut entendre les commentaires faits par des personnes en train de réaliser leurs actions, commentaires qui ne sont destinés à personne en particulier mais qui accompagnent l'accomplissement d'actions. En ce qui concerne la gestuelle, il s'agit dans ce cas de gestes réalisés intentionnellement pour communiquer de l'information.

c. Support à la collaboration

La conscience de groupe de l'espace de travail est utile pour différentes activités. Gutwin et Greenberg en mentionnent cinq en particulier (cf. Table 6).

Table 6: Activités dans lesquelles la conscience de groupe de l'espace de travail est utilisée (source : [83], p. 432)

Activité	Bénéfice de la conscience de groupe
Gestion du couplage	Assister les personnes pour gérer les transitions entre travail individuel et collectif.
Simplification de la communication	Permettre aux personnes d'utiliser l'espace de travail et les artefacts comme accessoires (<i>props</i>) de la conversation.
Coordination des actions	Assister les personnes dans la planification et l'exécution des actions pour qu'elles s'intègrent sans friction les unes avec les autres.
Anticipation	Permettre aux personnes de prédire les actions des autres et l'activité à différentes échelles de temps.
Assistance	Assister les personnes dans la compréhension du contexte où de l'aide est requise.

En premier lieu, elle facilite la gestion du couplage des activités. Le 'couplage' est défini comme '*le degré avec lequel les personnes sont en train de travailler ensemble*'. Ce concept peut s'exprimer en terme de '*quantité de travail qu'une personne peut réaliser avant de nécessiter une discussion, une instruction, une action, une information ou une consultation avec une autre personne*'. La conscience de groupe aide à déterminer le moment approprié pour initier un couplage plus intense sans interrompre les autres de manière impromptue et inadéquate.

Deuxièmement, elle permet de simplifier la communication verbale en offrant aux personnes la possibilité d'utiliser l'espace de travail et les artefacts qui s'y trouvent pour rendre leurs interactions plus efficaces. Néanmoins, la construction de telles communications non-verbales nécessite que l'expéditeur du message soit informé de ce que le destinataire peut voir. La conscience de groupe de l'espace de travail représente donc un élément du socle conversationnel commun (*conversational common ground*).

Ensuite, elle contribue à mettre en place une coordination des actions à accomplir dans le cadre d'une activité collaborative en informant les participants sur les limites temporelles et spatiales des actions des autres et en leur permettant d'insérer judicieusement leur prochaine action dans le flux global des opérations. La conscience de groupe de l'espace de travail est également utile lors des phases de planification et de re-planification des activités. En effet, les décisions dépendent alors de ce que les autres participants ont fait, de ce qu'il s'apprêtent à faire et de ce qu'il leur reste à faire.

La conscience de groupe de l'espace de travail facilite aussi le phénomène d'anticipation, c.-à-d. le fait que les personnes agissent sur base de leurs attentes ou de leurs prédictions de ce que les autres vont faire dans le futur. Les prédictions peuvent concerner une échelle de temps très courte et se baser sur les événements du passé immédiat ou une échelle plus longue et reposer sur des comportements répétitifs.

Enfin, la conscience de groupe de l'espace de travail aide les personnes à déterminer quelle assistance est requise par les autres et quelle assistance est appropriée à quel moment.

4. Discussion

Les différentes théories de l'activité coopérative présentées précédemment (cf. p. 13) nous permettent de mettre en évidence un premier concept central : la dichotomie entre un *monde interne* au cœur du cerveau humain et un *monde externe* observable. On peut en déduire qu'une activité coopérative présentera donc toujours une face visible, concrétisée par différents types d'interactions : entre *sujets* et *objets* par l'intermédiaire d'*outils* (cf. théorie de l'activité) ; entre *personnes* et *artefacts* (cf. cognition distribuée) ; entre *acteurs* et *ressources* (cf. théorie de la coordination) ou entre *individus* et *artefacts* (cf. ergonomie cognitive).



Fig. 8: L'iceberg de la coopération⁴

C'est cette partie émergée de l'iceberg de la coopération (cf. Fig. 8) que permet de supporter une plate-forme électronique. Il doit donc d'emblée être précisé que notre démarche se limitera à cet aspect visible de la coopération. Bien qu'elle présente un intérêt indéniable, l'étude des mécanismes internes au cerveau humain utilisés dans des activités coopératives ne figure pas au centre de nos préoccupations.

Par ailleurs, différentes théories de l'activité coopérative mettent en évidence la présence et la nécessité d'objets intermédiaires pour supporter la coopération. Ce rôle est joué par des *outils* dans la théorie de l'activité, par des *représentations externes* dans la théorie de la cognition distribuée ou par des *artefacts* dans l'ergonomie cognitive. Dans ce contexte, notre démarche consiste à nous focaliser sur un type bien particulier d'objets intermédiaires : les plates-formes électroniques de coopération. Bien que pleinement conscients du fait que seule une partie de la coopération visible est en pratique supportée par ces technologies informatiques, nous considérons néanmoins que, compte tenu de leur diffusion croissante, elles méritent un intérêt particulier.

⁴ Auteur de la photo-composition : Ralph Clevenger

Notre recherche se focalise spécifiquement sur les représentations graphiques d'informations relatives à des plates-formes électroniques de coopération. La théorie de la cognition distribuée nous fournit un cadre de réflexion particulièrement utile à ce niveau. En effet, les travaux de Zhang et Norman (1994) sur les rôles que peuvent jouer les représentations externes s'appliquent aisément aux techniques de visualisation d'information. Tout d'abord, l'aide à la mémorisation constitue sans aucun doute une des fonctions visées par les représentations graphiques que nous aurons à imaginer. La représentation de l'historique⁵ des événements associés à une plate-forme électronique illustre parfaitement ce rôle. Les représentations externes peuvent également être destinées à mettre à disposition des informations sous une forme directement perceptible et utilisable. Ici encore, nous observons un parallèle évident avec l'objectif premier de la discipline de la visualisation d'information : améliorer la cognition de données en utilisant les capacités de perception visuelle de l'être humain. Si les représentations graphiques que nous proposons s'avère difficiles à comprendre et dès lors peu utilisables, elles présenteront peu d'intérêt pour soutenir la coopération. Le concept d'*affordance*, devra lui aussi être intégré dans nos réflexions quand nous concevrons des techniques de visualisation. Rendre leur mode d'utilisation aussi intuitif et naturel que possible figurera sans nul doute dans la liste de nos objectifs.

La théorie de l'ergonomie cognitive offre un cadre théorique pour conceptualiser la différence entre tâche prescrite et activité effective. En mettant en évidence le caractère non complètement prescriptible et informel de beaucoup d'activités de travail, elle démontre l'intérêt d'étudier la visualisation des événements tels qu'ils se sont produits et de ne pas se limiter à la représentation de situations coopératives dans leur forme 'souhaitée'. Nous ne destinerons donc pas notre recherche prioritairement à la visualisation d'éléments tels qu'un plan de travail (*planning*) ou l'enchaînement pré-défini de tâches (*workflow*) mais plutôt à celle d'événements qui se sont effectivement produits.

Une plate-forme électronique de coopération étant par nature un outil de TCAO (CSCW), il est important de préciser quels types d'outils supportant la coopération nous considérerons dans notre recherche. De notre point de vue, le critère essentiel concerne la mise à disposition sur une plate-forme de coopération. Si on se réfère à la typologie de Johansen, une plate-forme électronique couvre habituellement les cases 'espace différent – temps différent'⁶ et 'espace différent – temps identique'⁷ de la matrice 'espace – temps'.

Notre étude bibliographique sur le travail coopératif assisté par ordinateur nous a fait prendre conscience du rôle central joué par le concept de conscience de groupe. Elle nous a permis mieux comprendre quelle information doit être communiquée, à quel moment et sous quelle forme pour supporter efficacement la naissance et le maintien d'une conscience de groupe par des moyens informatiques. Cependant, nous n'avons proposé à ce stade qu'une revue des principes théoriques sous-jacents à ce concept, sans détailler les moyens concrets par lesquels cette conscience de groupe peut être supportée. Ce point particulier sera discuté dans notre étude de l'état de l'art en matière de visualisation d'informations associées à un contexte collaboratif qui fait l'objet d'un chapitre ultérieur (cf. p. 87).

La coopération telle qu'elle se produit dans le monde réel met en jeu une multitude de mécanismes de communication, de coordination, de partage d'information, basés en autres sur la construction d'une conscience de groupe de cette situation particulière.

D'un point de vue générique, nous avons modélisé une situation de coopération comme un ensemble d'interactions de divers types entre des entités qui interagissent entre elles. Imaginons, pour illustrer notre réflexion, que trois types d'interactions soient présentes : les interactions de type A, B, et C (cf. Fig. 9). En pratique, ces types d'interaction peuvent représenter, par exemple, les échanges de courrier postal, les réunions en face à face, les conversations sur un forum de discussion, les réunions utilisant la vidéoconférence ou les

⁵ Exemples : liste de courriels, agenda avec réunions, liste des fichiers téléchargés...

⁶ Exemples : courriels, publication de documents téléchargeables

⁷ Exemple : messagerie instantanée, chat

dépôts de document dans un système de partage de documents. Dans certains cas, un ensemble d'outils informatiques de type TCAO (CSCW) sont utilisés pour faciliter cette coopération. Par exemple, on peut utiliser la messagerie électronique pour remplacer certains échanges de courrier par voie postale classique ou on peut substituer certaines réunions en face à face par des réunions à distance utilisant la vidéoconférence.

Quel que soit le contexte, il faut en permanence garder à l'esprit que seules certaines interactions sont concernées par ces outils informatiques. De plus, la gamme des interactions réelles qui sont supportées par des logiciels de type TCAO varie selon les contextes. Si nous nous référons à notre modèle, on pourrait imaginer une situation 1 dans laquelle seules les interactions de type A sont supportées par des outils informatiques et une situation 2 où les interactions de type A et C sont supportées (cf. Fig. 9). Ces deux situations diffèrent naturellement quant à l'importance des outils informatiques dans le support de la coopération. On voit donc émerger le concept de 'couplage' entre la coopération complexe du monde réel et la partie de celle-ci qui est gérée par des outils informatiques du type TCAO. Nous définirons le **couplage** comme **la proportion des interactions réelles associées à une activité coopérative qui sont mises en œuvre via l'utilisation d'outils informatiques**.

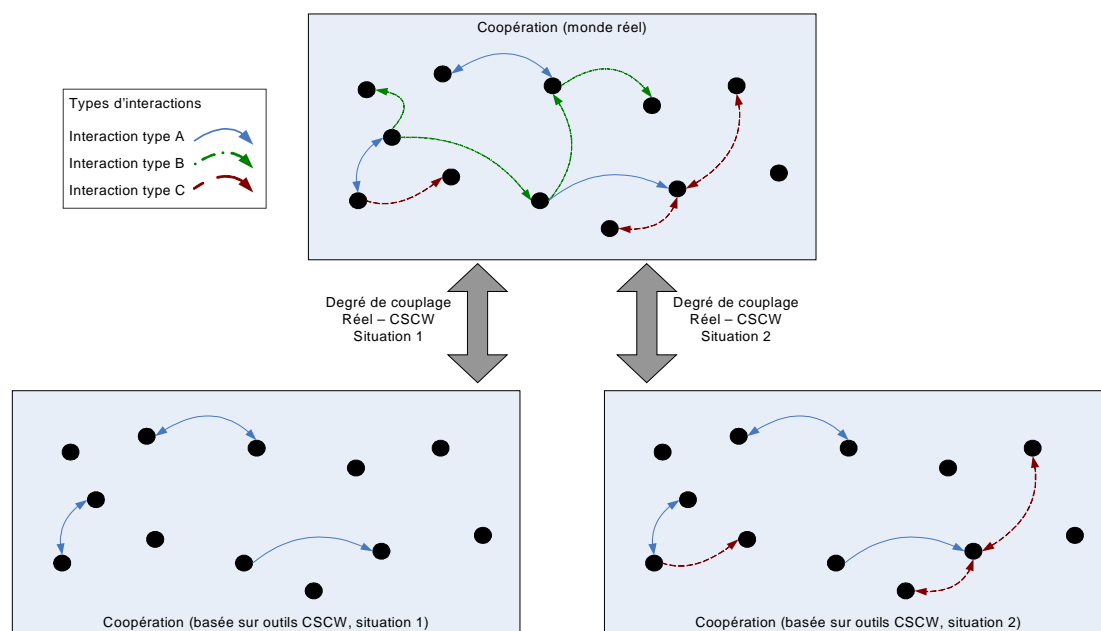


Fig. 9: Illustration du concept de couplage entre les coopérations informatisée et réelle

On peut rencontrer en pratique une large gamme de situations allant d'un couplage fort, dans lesquelles la quasi-totalité des interactions fait appel aux outils informatiques, à un couplage faible ou nul dans lesquelles l'informatique n'est pratiquement pas utilisée pour supporter la coopération. La rédaction coopérative de l'encyclopédie en ligne 'wikipedia' ou le développement de logiciels libres via les plates-formes de type 'Sourceforge' sont deux exemples de couplage fort. Par contre, l'élaboration d'une stratégie d'entreprise au cours d'un séminaire résidentiel ou la construction d'une habitation unifamiliale par une PME illustrent plutôt des situations de couplage faible dans lesquelles la majorité des interactions se produisent en présence physique des acteurs concernés.

A priori, différentes propriétés des entités et des interactions peuvent influencer le niveau de couplage. On peut ainsi raisonnablement considérer que des facteurs tels que le profil individuel des personnes⁸, le contexte⁹, le degré de formalisme exigé¹⁰ ou encore le niveau de

⁸ Niveau de formation, activité professionnelle, âge, expérience avec l'informatique...

complexité des informations échangées¹¹ jouent un rôle dans le choix des outils retenus pour instancier une interaction dans une situation de coopération.

Cette problématique est étudiée particulièrement par la communauté de recherche sur la communication médiatisée par l'ordinateur (*Computer-Mediated Communication*). Nous mentionnerons ici une des théories les plus connues de ce domaine : la Théorie de la Richesse des Médias (*Media Richness Theory*) proposée par Daft et Lengel (1984 ; 1986) dans les années 80 et qui fut l'objet de divers travaux par la suite (p.ex. Newberry, 2001 pour une application au domaine du *e-learning*). Cette approche défend l'idée que les différents médias de communication possèdent une certaine richesse, en termes de capacité à résoudre les ambiguïtés, à négocier des interprétations variables et à faciliter la compréhension. En d'autres termes, chaque médium possède une capacité propre à transmettre des informations brutes (*data*) et des méta-informations sur celles-ci, comme par exemple des informations sur les personnes en train de communiquer. Dans cette approche, par exemple, une conversation sur un forum de discussion ou une note mémo sont plutôt qualifiés de médias pauvres alors qu'une communication en face à face ou une conversation par vidéoconférence sont associés à un niveau élevé de richesse¹². La théorie indique également que, selon les circonstances et le type d'information à faire passer, les interlocuteurs opteront en priorité pour des médias plus riches ou plus pauvres, ce qui est également résumé dans le concept de concordance entre la tâche à accomplir et le médium de communication adéquat (*Task/Media Fit*).

La théorie de la richesse des médias a une vocation explicative, dans le sens où elle vise à mettre en évidence pourquoi certains médias sont utilisés préférentiellement dans certaines situations. Par contre, notre concept de couplage vise essentiellement à préciser la portée des propositions qui seront développées dans la suite de nos travaux. Pour nous, il s'agit principalement d'attirer l'attention sur le fait que la partie électronique de la coopération sur laquelle se focalisent nos travaux ne représente qu'une fraction de la coopération réelle.

Nous pouvons à présent préciser l'espace sur lequel nous travaillons et les hypothèses sous-jacentes à notre approche. Nous cherchons à étudier **des situations dans lesquelles un certain niveau de coopération entre des personnes est partiellement ou totalement supporté par des outils informatiques**.

Notre recherche s'appliquera donc, par exemple, à des situations dans lesquelles une plateforme électronique de coopération est utilisée pour gérer un consortium de recherche international, une grappe d'entreprises actives dans un secteur industriel particulier ou un projet de construction.

D'un point de vue général, notre réflexion repose sur la modélisation de la coopération sous la forme d'interactions entre des entités (cf. Fig. 10). De notre point de vue, une activité coopérative sera donc fondamentalement composée d'*entités*, qui peuvent être des personnes ou des ressources, et d'interactions entre ces entités, définies comme '*tout échange d'information entre des entités*'.

Dans une section précédente, nous avons expliqué que, dans la réalité, toutes les interactions ne sont pas réalisées en utilisant des outils informatiques (cf. iceberg de la coopération). Nous avons donc défini le concept plus restrictif d'*interaction informatisée*, définie comme '*tout échange d'information entre des entités qui est basé sur l'utilisation d'outils informatiques*'. L'échange d'information peut se produire entre des entités de type '*personne*'. C'est le cas,

⁹ Relations professionnelles, relations entre membres d'une famille, relations entre amis...

¹⁰ Communication officielle avec une administration, échanges informels entre collègues, discussion entre étudiants...

¹¹ Invitation à une réunion, discussion stratégique sur l'avenir de l'entreprise, prise d'avis dans une négociation délicate...

¹² Notons dans ce contexte que les travaux du laboratoire LUCID de l'Université de Liège indiquent que, dans un processus de conception en architecture ou ingénierie, une esquisse doit plutôt être considérée comme un médium riche.

par exemple, d'un message électronique ou d'une réunion par vidéoconférence. Il peut aussi concerner des entités de type '*personne*' et d'autres de type '*ressource*'. Par exemple, on peut citer le dépôt d'un document dans un système partagé.

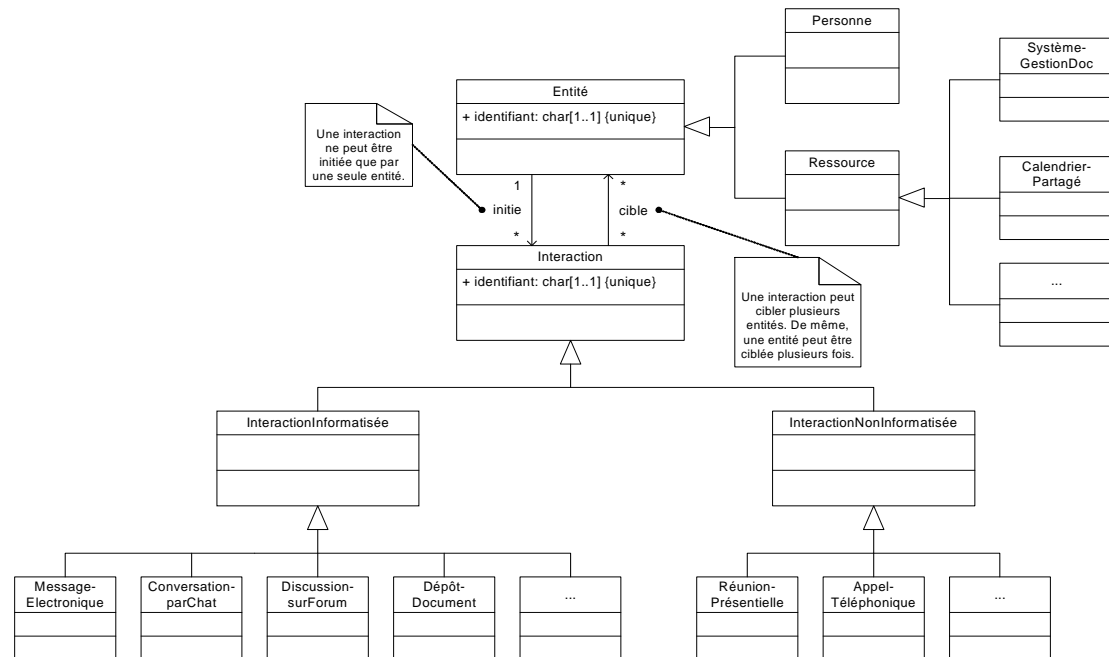


Fig. 10: Modélisation de la coopération sous forme d'entités en interactions

En conclusion, notre recherche n'a pas pour ambition de proposer des visualisations holistiques de situations de coopération car une telle entreprise nous semble exagérément ambitieuse d'un point de vue conceptuel et impossible à mettre en œuvre d'un point de vue pratique. Nous posons donc comme hypothèse de départ que tant les mécanismes internes de la cognition humaine que les modalités de coopération sans trace électronique sont exclues de notre travail. Nous nous focaliserons de plus sur la représentation graphique des activités réelles plutôt que sur celle des tâches prescrites.

Chapitre 3 : Visualisation d'information

« On ne connaît pas complètement une science tant qu'on en sait pas l'histoire. »

A. Comte

1. Introduction générale

A l'échelle de l'histoire des sciences, l'idée de visualiser des données par le biais de graphiques est relativement récente. Elle date en fait de la fin du 18^{ème} siècle avec les travaux précurseurs de William Playfair dans le domaine des sciences économiques (cf. Hollan et al., 1997 ; Tufte, 2001). Depuis, l'utilisation de représentations graphiques des informations a percolé dans la quasi totalité des disciplines scientifiques : biologie, physique, histoire, informatique...

Avant tout, il est indispensable de préciser la nature du concept de 'visualisation graphique d'information'. Les travaux fondamentaux de Jacques Bertin constituent une référence incontournable à ce niveau. Bertin (1967) définit la représentation graphique comme étant *'la transcription, dans le système graphique de signes, d'une pensée, d'une « information » connue par l'intermédiaire d'un système de signes quelconques.'* Il insiste également sur la nécessité de distinguer la graphique et le graphisme (2000). *'La graphique utilise les propriétés de l'image visuelle pour faire apparaître les relations de ressemblance et d'ordre entre les données. [...] Elle s'applique à un ensemble préalablement défini : le tableau des données et constitue ainsi la partie rationnelle du monde des images dans le classement logique des systèmes de signes fondamentaux. [...] Il faut éviter la confusion entre la graphique, outil de travail, qui ne traite que d'ensembles définis par avance (le tableau des données) [...] et le graphisme – figuratif ou non – qui, au contraire, cherche à définir un ensemble dans l'esprit de l'observateur. Le graphisme est un art, libre mais toujours discutable.'* En d'autres termes et pour préciser encore la portée de la graphique, Bertin ajoute : *'La graphique poursuit deux objectifs :*

- *traiter les données pour comprendre et en tirer l'information ;*
- *communiquer s'il y a lieu cette information ou un inventaire de données élémentaires.'*

Dans l'introduction d'un des ses ouvrages de référence : *'The Visual Display of Quantitative Information'*, Tufte (2001) exprime lui aussi une idée similaire : *'L'excellence en matière de graphes statistiques consiste à communiquer des idées complexes avec clarté, précision et efficacité.'* Schneiderman, un autre auteur incontournable de ce domaine, définit la discipline de la visualisation d'information comme *'l'utilisation de représentations visuelles interactives de données abstraites destinée à amplifier la cognition'* (Schneiderman et Plaisant, 2005).

Card (2003) propose un positionnement global de cette discipline en relation avec d'autres domaines scientifiques. *'La visualisation d'information doit être distinguée des domaines qui y sont liés : la visualisation scientifique est comme la visualisation d'information, mais appliquée aux données scientifiques, typiquement basées sur des aspects physiques. [...] La visualisation d'information et la visualisation scientifique appartiennent au champ plus large de la graphique des données (data graphics), qui est l'utilisation de représentations visuelles abstraites non-représentationnelles pour amplifier la cognition. La graphique des données fait partie du design informationnel (information design), qui est concerné par les représentations externes pour amplifier la cognition. Au niveau le plus haut, nous pourrions considérer le design informationnel comme une partie de la cognition externe, qui concerne les utilisations du monde extérieur pour accomplir des processus cognitifs.'*

Une représentation graphique peut être assimilée à un modèle d'une réalité plus ou moins complexe qui permet de mettre en évidence de manière visuelle certaines propriétés des données sous-jacentes. Différents auteurs rejoignent cette position.

- *'La visualisation de l'information s'attache à donner une représentation graphique à des entités ne possédant pas nécessairement une transcription visuelle directe.'* (Bruley, 1999)
- *'La visualisation offre une méthode pour voir l'invisible.'* (Shneiderman, 1998)
- *'Les interfaces graphiques peuvent fournir un moyen de voir de l'information qui est cachée ou inaccessible dans une représentation textuelle.'* (Donath et al., 1999)
- *'La visualisation d'informations abstraites a la puissance de révéler des motifs, des grappes, des fossés ou des observations atypiques dans des données statistiques, des cours sur les marchés financiers, des répertoires d'ordinateurs ou des collections de documents.'* (Shneiderman, 1998)
- *'Une des propriétés de la représentation visuelle est d'offrir à l'utilisateur une vue d'ensemble de l'espace informationnel.'* (Chasseur, 2002)

Bertin (cité dans Novick et Hurley, 2001) complète ce point de vue en notant qu'une représentation graphique adéquate de données fournit de l'information à la fois sur l'organisation globale de celles-ci et sur les éléments individuels.

Par ailleurs, de multiples travaux de recherche ont établi que le mode de représentation graphique constitue un outil de travail très puissant.

- Les graphiques permettent de mieux comprendre l'état d'une situation et facilitent la résolution de problème (Larkin et Simon, 1987).
- Les graphiques sont particulièrement adaptés quand les utilisateurs doivent analyser et comparer un ensemble de données (Smith et Mosier, 1986).
- Les graphiques sont très utiles à la visualisation et la recherche d'information (Robertson et al., 1993).
- La présentation d'une information visuelle augmente la compréhension et la rétention des données (Unamath et Scamell, 1988).
- Les graphiques aident à la reconnaissance et réduisent les traitements en mémoire de travail (Lohse, 1995).

Etant acquis que les représentations graphiques peuvent être d'une grande utilité, il est naturel de s'intéresser à ce qui influence la qualité de celles-ci. Trois auteurs nous offrent leur vision de cette problématique.

Selon Bertin (1998), l'efficacité d'une construction graphique est définie par la proposition suivante. *'Si, pour obtenir une réponse correcte et complète à une question donnée et toutes choses égales, une construction requiert un temps de perception plus court qu'un autre construction, on dira qu'elle est plus efficace pour cette question.'*

Lohse (1997) affirme qu'une visualisation d'information efficace transforme la structure de coût de l'information, par exemple en remplaçant un travail cognitif d'inférence sur les données par une perception directe.

Tufte (2001) plaide en faveur de représentations graphiques simples, qui minimisent l'utilisation de l'encre destinée à dessiner tout autre élément que les données. Son principe de base est le suivant : *'Avant tout autre chose, montrez les données'*.

Nous voyons donc poindre ici une idée essentielle dans le domaine de la visualisation de l'information : toutes les représentations graphiques ne sont pas équivalentes pour illustrer un jeu de données. En fait, choisir la représentation graphique la plus adéquate pour illustrer une (ou plusieurs) propriété(s) d'une ensemble de données constitue une des tâches les plus délicates de la discipline de la visualisation d'information. Comme le fait remarquer Lohse (1997), *'bien que les ordinateurs ait automatisé le mécanisme de production d'un graphe, ils n'ont pas automatisé la tâche de choisir le bon type de graphe qui illustre le mieux l'information'*. Il insiste en affirmant : *'alors qu'il est facile d'inventer un nouveau type de graphe, il est difficile d'en inventer un qui marche bien'*. Hollan et al. (1997) rejoignent cette

position en affirmant : ‘des visualisations bien conçues peuvent être énormément utiles mais les créer reste un véritable défi’.

La discipline de la visualisation d'information s'avère souvent peu connue et dès lors ses éléments constitutifs sont souvent mal délimités. Il nous a dès lors semblé utile d'en expliquer les principes élémentaires à partir de propositions de deux auteurs de référence dans le domaine : Stuard Card et Robert Spence.

Card (2003) a proposé un modèle de référence (cf. Fig. 11) permettant de bien différencier les différentes opérations associées à la visualisation d'information.

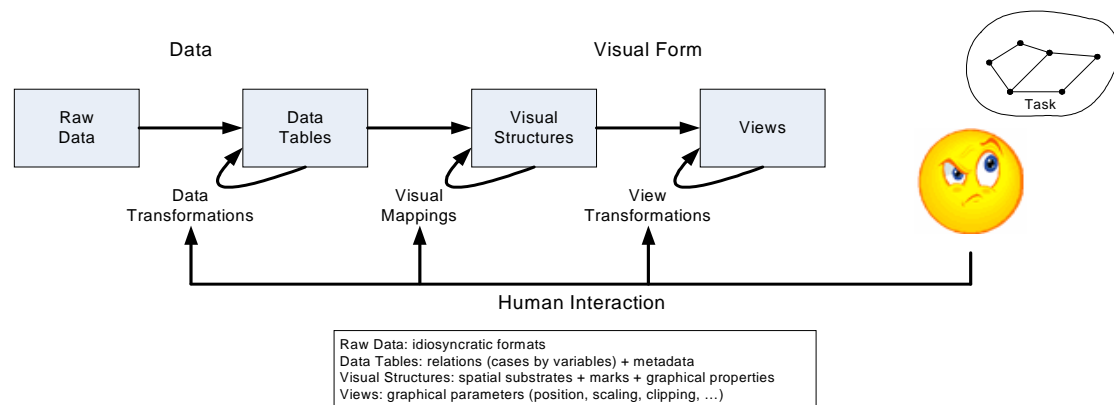


Fig. 11: Modèle de référence de Card (adapté de [37], p. 554)

La matière première de la visualisation d'information se compose de données brutes (*raw data*) qui peut se présenter sous la forme de texte, de chaînes de caractères indexées ou encore de tableaux. Ces données brutes sont transformées sous la forme tables de données avec des méthodes telles que, par exemple, le *multidimensional scaling*. Les tables de données sont en fait des descriptions canoniques des données dans un format de type : variables x cas étendu pour inclure des métadonnées. La transformation suivante (*visual mapping*) est la plus importante. Elle produit des structures visuelles qui expriment les valeurs des variables pour les différents cas au moyen d'un vocabulaire d'éléments visuels : le positionnement spatial, les signes et les propriétés graphiques. Finalement, les structures visuelles peuvent être encore modifiées par des transformations de vue (*view transformations*), telles que des distorsions géométriques. Les vues sont le résultat du processus et sont présentées à l'utilisateur.

L'interaction de l'utilisateur est un autre élément fondamental du modèle. Elle peut porter sur chacune des modélisations de la réalité présentes dans le processus : les données brutes, les tables de données, les structures visuelles et les vues. A titre d'exemple, cette composante du modèle permet de préciser à quel niveau agit un opérateur de filtrage des informations présent dans une interface graphique.

Spence (2007) propose également une vue synthétique de la visualisation d'information (cf. Fig. 12). Il rappelle que *'la visualisation d'information est essentiellement la formation d'un modèle mental des données'*.

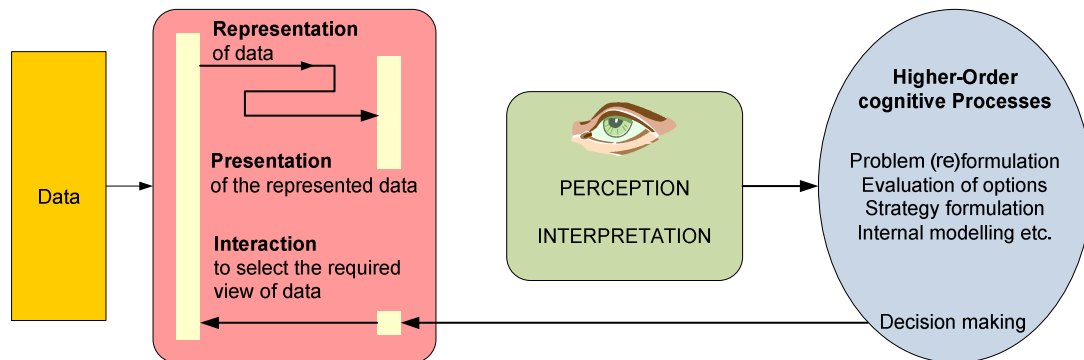


Fig. 12: Éléments constitutifs de la visualisation d'information (adapté de [190], p. 26)

Les données constituent donc le point de départ de tout le processus. Dans un premier temps, une représentation (modèle) des données est élaborée en vue de leur présentation (graphique). Ensuite, viennent les phases qui mettent en œuvre les capacités cognitives de l'observateur de la représentation. La simple perception du graphique est suivie de son interprétation. Des processus cognitifs de plus haut rang sont alors déclenchés, tels qu'une reformulation du problème, une évaluation des options, la formulation d'une stratégie. Une décision en résulte qui conduit dans la plupart des cas à une interaction pour affiner certains paramètres de sélection des données ou de leurs propriétés.

Nous concluons cette brève introduction au domaine de la visualisation d'information en rappelant deux principes fondateurs de cette discipline :

- La visualisation d'information a pour but d'étendre les capacités cognitives de l'être humain.
- Le défi de la visualisation d'information consiste dans la mise en relation adéquate de données issues d'un traitement sur des informations brutes et de représentations visuelles éventuellement interactives.

2. Taxonomies des techniques de visualisation

2.1 Perspective historique

En 1969, H. Simon affirmait dans son ouvrage intitulé 'Les sciences de l'artificiel', devenu depuis un classique : *'un premier pas vers la compréhension d'un ensemble quelconque de phénomènes est de développer une taxonomie'*. En 1990, lors de la première conférence IEEE sur la Visualisation, organisée à San Francisco, Lohse et ses collègues mettaient en évidence l'importance d'une classification : *'La classification est un processus fondamental de la science. Les classifications identifient les concepts nécessaires pour un développement de théories dans un domaine scientifique. Toutes les sciences sont construites sur des classifications qui structurent leurs domaines de recherche.'* Au cours de la même conférence, Wehrend et Lewis (1990) déclaraient : *'Les utilisateurs de visualisations dans différentes disciplines d'application manquent d'un cadre conceptuel commun dans lequel les techniques peuvent être décrites et partagées.'* Lors d'un workshop tenu en 1991 sur les environnements de visualisation scientifique, un groupe de chercheurs élaborait une liste de domaines de recherches prometteurs (Butler et Hansen, 1992). Parmi celle-ci, figurait *'la construction d'une taxonomie complète et détaillée des données [à visualiser]'*. Un an plus tard, en 1992, Gruia-Catalin et Cox déclaraient, en proposant une taxonomie des visualisations de programmes informatiques : *'la taxonomie proposée devient le véhicule avec lequel nous réalisons une revue systématique des systèmes, des techniques, des tendances, et des idées actuelles [...]'*. En 1994, Lohse et ses collègues réaffirmaient dans un article proposant une classification des informations visuelles : *'la classification se trouve au cœur de chaque discipline scientifique. Les classifications structurent les domaines d'investigation systématique et fournissent des concepts pour développer des théories permettant d'identifier les anomalies et de prédire les futurs besoins de recherche'*. Ils déclaraient dans le même esprit : *'Si la visualisation veut continuer à avancer comme une science interdisciplinaire, elle doit devenir plus qu'un fourre-tout de techniques pour afficher des données'*. Trois ans plus tard, en 1997, Card et Mackinlay observaient : *'Il est maintenant approprié de commencer à décrire et à analyser des portions de l'espace de conception [en visualisation d'information] de manière à comprendre les différences entre les représentations (designs) et de suggérer de nouvelles possibilités.'* Quelques années plus tard, Baudel (2002a) mettait en avant *'la nécessité d'aborder de manière générique la construction de techniques de visualisation, de proposer une taxonomie [...]'*. En 2003, Pfitzner, Hobbs et Powers constataient *'le manque d'analyse paradigmatique et de caractérisation des types de techniques de visualisation d'information'*.

Nous pouvons donc conclure que, comme toute autre discipline scientifique, la visualisation d'information a un besoin critique de taxonomies. Plusieurs auteurs ont donc présenté des propositions pour répondre à ce besoin. Nous présentons une revue des plus importantes d'entre elles dans la section suivante en adoptant un ordre chronologique.

2.2 Taxonomies proposées dans la littérature

2.2.1. Bertin (1973, 1998 & 2000)

Bertin (1998, 2000) a formalisé dès 1967 sa théorie de la sémiologie graphique. Il est considéré comme un des pères fondateurs de l'étude des représentations graphiques d'information. La graphique *'utilise les propriétés de l'image visuelle pour faire apparaître les relations de différence, d'ordre ou de proportionnalité entre des données'*. Elle s'applique à un ensemble préalablement défini, appelé le tableau de données. La graphique poursuit deux objectifs :

- *'Traiter les données pour comprendre et en tirer l'information'*
- *'Communiquer s'il y a lieu, cette information ou un inventaire de données élémentaires'*

Bertin propose une classification de l'information à représenter, puis une classification des propriétés de la représentation graphique.

Bertin précise tout d'abord en quoi consiste une information. '*Dans une représentation graphique, on appellera information le contenu traductible d'une pensée. Il est constitué essentiellement par une ou plusieurs correspondances originales entre un ensemble fini de concepts de variation et un invariant.*' (1998, p. 9) Il appelle '*composantes*' les concepts de variation mis en œuvre (p.ex. information à deux composantes : variation de coût, variation du temps).

Bertin identifie trois éléments pour analyser une information :

- **le nombre de composantes** ;
- **la longueur de chaque composante**, définie comme le nombre de 'catégories' ou 'éléments' qu'elle permet d'identifier ;
- **le niveau d'organisation** entre les composantes ou entre les catégories :
 - *Niveau qualitatif*, qui groupe tous les concepts de simple différenciation (p.ex. métiers, produits...) ;
 - *Niveau de l'ordre*, qui groupe tous les concepts susceptibles d'ordonner les éléments d'une manière universellement admise ;
 - *Niveau quantitatif ou métrique*, qui nécessite la présence d'une unité comptable.

Ensuite, il s'intéresse aux propriétés de la représentation graphique. Il appelle '*variables visuelles*' les composantes du système graphique. Bertin considère que celui-ci dispose de huit variables visuelles regroupées en deux types : les variables du plan et les variables rétinienne (ou variables d'élévation).

- **Variables du plan**, appelées '*dimensions du plan*'.
- **Variables rétinienne** (ou variables d'élévation) : la taille, la valeur (niveau de gris), le grain, la couleur, l'orientation et la forme.

Par ailleurs, Bertin identifie également trois objets graphiques élémentaires :

- le **point** (position sans surface) ;
- la **ligne** (position linéaire sans surface) ;
- la **zone** (surface).

Bertin distingue les représentations graphiques en se basant sur le concept d'*imposition*, qui correspond à l'utilisation des deux dimensions du plan.

Dans un premier temps, il identifie quatre '*groupes d'imposition*', sur base des correspondances que les représentations graphiques établissent dans le plan :

- **Diagrammes** : établissent des correspondances entre tous les éléments d'une composante et tous les éléments d'une autre composante ;
exemple : variation du cours d'une action au cours du temps
- **Réseaux** : établissent des correspondances entre tous les éléments d'une même composante ;
exemple : relations entre individus autour d'une table
- **Cartes géographiques** : établissent des correspondances entre tous les éléments d'une même composante géographique inscrits dans le plan suivant la distribution géographique observée.
- **Symbolique** : établit des correspondances entre un seul élément et le lecteur. La correspondance est extérieure à la représentation graphique.
exemple : signal routier

Ensuite, pour les diagrammes et les réseaux uniquement, il distingue des *types d'imposition*, selon la disposition des composants :

- **Semis, Arbre**
Les dimensions du plan ne sont pas associées à une composante.
- **Rectiligne**
 - une dimension du plan est mobilisée d'une manière homogène, les catégories se suivent une fois pour toutes.
 - une dimension du plan est mobilisée d'une manière hétérogène, les catégories se répètent n fois.
 - une dimension du plan représente des quantités cumulées.
- **Orthogonale**
Chacune des deux dimensions orthogonales du plan représente une composante.
- **Circulaire**
Une composante est représentée par l'angle entre des rayons d'un cercle.
- **Polaire**
Une composante est représentée par la longueur des rayons et l'autre par l'angle entre eux.

La combinaison des types d'imposition et des variables d'élévation forme les *types de construction*. Il faut noter que la problématique du choix de la construction se pose pour les diagrammes et les réseaux.

Un des principes de base de Bertin est qu'une représentation graphique constitue toujours une réponse à une question, consciente ou non. Il s'est donc logiquement intéressé à la classification des questions possibles. Il propose une approche en deux temps.

- Tout d'abord il indique qu'il y a autant de **types de questions** que de composantes dans une information. Chaque composante fournit un type de question.
- Ensuite, il distingue 3 **niveaux de lecture** :
 - *Niveau élémentaire de lecture* : questions introduites par un seul élément de la composante et aboutissant à une seule correspondance.
 - *Niveaux moyens de lecture* : questions introduites par un groupe d'éléments de la composante (ces questions tendent à réduire la longueur des composantes)
 - *Niveau d'ensemble* : question introduite par l'ensemble de la composante (cette question tend à réduire toute l'information à une unique relation d'ordre entre les composantes).

Globalement, Bertin identifie 3 fonctions de la représentation graphique :

- **Enregistrer l'information** : créer une mémoire artificielle qui évite l'effort de mémorisation. La construction sera exhaustive mais peut-être non mémorisable dans son ensemble. La représentation graphique est un *inventaire*.
- **Communiquer l'information** : créer une image mémorisable qui inscrira l'information dans la mémoire. La construction sera mémorisable mais peut être non exhaustive. La représentation graphique est un *message*.
- **Traiter l'information** : fournir les dessins qui permettent procéder à la simplification et de la justifier. La construction sera mémorisable (pour les comparaisons) et exhaustive (pour les choix). La représentation graphique est un *instrument expérimental*.

Il est important de noter que Bertin étudie la construction de représentations graphiques statiques à deux dimensions. Il ne s'intéresse qu'à ce qui est représentable '*sur une feuille plane de papier blanc d'un format moyen, sous un éclairage normal*'. Il exclut explicitement les représentations graphiques incluant du mouvement. Il exclut également les représentations tridimensionnelles.

Bertin indique que le choix d'une représentation graphique doit suivre la règle suivante. '*Chaque variable visuelle a ses propriétés particulières de niveau et de longueur. Il importe que chaque composante soit transcrite par une variable ayant au moins le niveau et la longueur correspondants.*'

Bertin s'est également intéressé à l'efficacité d'une construction graphique qui est, selon lui, lié au temps de perception nécessaire pour obtenir une réponse correcte et complète à une question donnée. Sur cette base, il indique comment obtenir une construction efficace. Dans ce but, il définit le concept d'**image**, comme la '*forme visuelle significative perceptible dans l'instant minimum de vision*'. Le concept d'image correspond en fait à celui de '*forme*', '*pattern*' ou '*Gestalt*'. L'image ne doit pas être confondue avec la figure qui est l'unité apparente et illusoire définie par la feuille de papier, par un encadrement linéaire ou par un cadre géographique. Il conclut : '*les constructions les plus efficaces sont celles dans lesquelles toute question, quel qu'en soit le type ou le niveau, obtient une réponse dans l'exercice d'un seul instant de perception, une réponse perceptible en une seule image.*'

2.2.2. Roth et Mattis (1990)

La proposition de Roth et Mattis (1990) se situe dans le cadre d'un système de présentation automatique (*automatic presentation system*), défini comme un '*composant d'interface intelligent qui reçoit de l'information d'un utilisateur ou d'un programme et qui conçoit une combinaison de graphiques et de texte qui la communique de manière efficace*'.

Leur taxonomie vise principalement les données, exprimées sous la forme de n-uples 'objet-attribut-valeur' (p.ex. 'activité de conception – coût – 3000\$'), où les objets et les valeurs correspondent aux éléments d'un ensemble et les attributs à des relations.

Il s'agit d'une taxonomie portant sur les **caractéristiques de l'information** qui devraient être fournies à des êtres humains ou à des ordinateurs pour leur permettre de créer des présentations efficaces, de manière indépendante d'une application informatique. Cette caractérisation des données doit être comprise ici comme '*une description des propriétés sémantiques et structurales de l'information qui sont pertinentes pour la conception de présentations*'.

Roth et Mattis expliquent qu'ils se sont basés sur quatre groupes de critères pour juger de la pertinence des caractéristiques retenues :

- les caractéristiques qui sont nécessaires pour distinguer les types d'information que chaque technique graphique peut représenter ;
- les caractéristiques qui aident à ordonner les techniques graphiques, sur base de leur efficacité à communiquer différentes informations ;
- les caractéristiques qui peuvent être utilisées pour déterminer comment l'information peut être intégrée dans un moyen d'affichage ;
- les caractéristiques qui peuvent être utilisées facilement par des utilisateurs.

Ces auteurs identifient cinq dimensions permettant de caractériser les données à visualiser. Quatre d'entre elles sont statiques : les types de données (*Data Types*), les propriétés de structure relationnelle (*Relational Structure*), les relations entre les relations (*Relationships among Relations*) et l'arité (*Arity*). La cinquième fait référence aux buts de l'utilisateur (*User Information Seeking Goals*).

En ce qui concerne les **types de données**, ils précisent : '*Les blocs constitutifs d'une présentation graphique sont les techniques pour identifier un élément déterminé dans un ensemble de valeurs possibles. [...] Différentes caractéristiques d'un ensemble sont pertinentes pour choisir une technique graphique appropriée.*'

Tout d'abord, il faut s'intéresser à la présence d'une **relation d'ordre** entre les éléments de l'ensemble.

- Dans les ensembles quantitatifs, les éléments sont ordonnés numériquement. Roth et Mattis suggèrent que les techniques graphiques qui incluent une variation quantitative d'une dimension visuelle sont efficaces pour représenter de tels ensembles.
- Dans les ensembles ordinaux, la relation d'ordre est particulière selon la sémantique de l'ensemble. La représentation de ces ensembles requiert des techniques qui peuvent énumérer explicitement chaque élément selon un axe ou une clé. Les valeurs intermédiaires ne peuvent pas être interpolées.
- Dans les ensembles nominaux, les éléments ne sont pas ordonnés. Roth et Mattis déclarent qu'ils peuvent être représentés efficacement avec des couleurs ou des formes, qui ne varient pas quantitativement, pour autant que la taille de l'ensemble soit petite.

Ensuite, ces auteurs mettent en évidence la **distinction entre 'coordonnées' et 'quantités'**. Ils définissent des coordonnées comme des éléments d'un ensemble ordonné qui spécifient un point ou une localisation temporellement, spatialement ou relativement à un autre repère (ex. time-zone, latitude...). Par contre, les quantités ne sont pas insérées dans des références particulières (ex. poids, nombre de jours...)

Enfin, Roth et Mattis mentionnent **l'appartenance à un domaine particulier** de l'ensemble à représenter. Ils citent, par exemple, le domaine des données temporelles, spatiales, des données de température ou de masse.

En second lieu, ils étudient les **propriétés de structure relationnelle**. Il s'agit de la manière dont des relations mettent en correspondance des éléments d'un ensemble avec un autre. Roth et Mattis identifient trois propriétés importantes des relations dans ce contexte.

- **La couverture relationnelle** (*Relational Coverage*) exprime le fait que chaque élément d'un ensemble puisse être mis en correspondance avec au moins un élément d'un autre ensemble. Plusieurs types de non-couverture sont identifiés :
 - Données manquantes (*missing data*)
 - Relation non applicable à certains éléments (*not-applicable relation*)
 - L'absence de valeur est porteuse d'information (*informative no-value*)

Les diverses représentations graphiques possèdent une capacité très variable à gérer les différents types de non-couverture relationnelle.

- **La cardinalité** (*Cardinality*) exprime le nombre d'éléments d'un ensemble vers lesquels une relation peut pointer à partir d'un élément d'un autre ensemble. Cette notion fait référence au nombre de valeurs que peut prendre un attribut d'un objet : une seule valeur (*single-valued*), plusieurs valeurs fixes (*fixed-multiple-valued*), un nombre variable de valeurs (*variably-valued*).
- **L'unicité** (*Uniqueness*) indique que chaque élément d'un ensemble est associé à une valeur spécifique unique, c.à.d. une valeur que lui seul peut prendre.

Troisièmement, Roth et Mattis abordent la question de **l'expression des relations entre les relations** sous deux aspects.

- Le premier fait référence aux difficultés posées par la représentation de relations complexes (c.-à-d. d'arité supérieure à 2). Ils proposent d'utiliser des **types de données complexes** (*Complex Data Type*) qui définissent des rôles permettant de caractériser les relations entre des relations binaires simples ou des arguments de relations complexes (ex : intervalles, moyenne, coordonnées 2-D...). Par exemple, la relation ternaire 'période d'emploi' met en relation des employés, une date de début et une date de fin. Même si on transforme cette relation ternaire en deux relations binaires 'début d'emploi' et 'fin d'emploi' qui associent chacune des employés et des dates, la représentation graphique doit tenir compte des relations sémantiques existant entre ces relations binaires.

- Le second aspect concerne les *dépendances algébriques* qui peuvent exister entre des attributs (relations) ou entre des valeurs au sein d'un ensemble de données. Par exemple, le 'budget total' représente la somme de plusieurs 'budgets spécifiques'. Une bonne représentation graphique doit intégrer cet élément.

En quatrième lieu, Roth et Mattis se concentrent sur l'**arité des relations**. Ils insistent sur le fait de distinguer les *relations unaires* d'une part et les *relations d'arité supérieure ou égale à deux d'autre part*. Les relations visant à distinguer un sous-ensemble d'entités ayant une propriété particulière au sein d'un ensemble plus vaste sont des relations unaires dont la nature spécifique doit être exprimée lors de leur représentation graphique.

En ce qui concerne les **buts de l'utilisateur**, il s'agit ici de prendre en compte les buts de l'utilisateur (ou de l'application) pour lequel des données sont visualisées. Roth et Mattis distinguent des buts associés à la *fonction* de présentation des données et d'autres associés à la *distribution* des données dans la présentation.

- En ce qui concerne la *fonction de présentation*, il faut distinguer différents buts essentiels :
 - Recherche d'une valeur précise
 - Comparaison de valeurs au sein de relations
 - Comparaison de relations deux à deux ou n à n pour un même ensemble de données
 - Distribution de valeurs pour une relation
 - Corrélations fonctionnelles entre attributs
 - Indexation d'ensembles de données au sein d'une relation
- Au niveau de la *distribution de l'information* dans la présentation, les buts de l'utilisateur déterminent l'apparement entre les relations et dès lors influencent comment elles doivent être représentées (p.ex. via des zones d'affichage différentes).

En conclusion, nous pouvons dire que la taxonomie de Roth et Mattis porte essentiellement sur les caractéristiques de l'information à visualiser. Par contre, elle ne propose pas de classification des techniques de représentation graphique. Elle utilise seulement des exemples de telles techniques pour illustrer sa taxonomie des données.

Il s'agit donc clairement d'une proposition de taxonomie des données. Les auteurs ajoutent également une classification des buts de l'utilisateur mais qui n'est pas vraiment mise en relation avec la classification des données.

2.2.3. Robertson (1990)

Robertson (1990) a constaté que le choix des représentations graphiques se fait souvent de manière ad hoc sur base des conseils d'un 'expert en visualisation'. Pour rationaliser ce processus, il met en évidence le besoin de développer une méthodologie formalisée et solide permettant à un être humain ou à un ordinateur de choisir judicieusement des représentations de données adéquates.

Sa démarche se décompose en trois étapes :

- établir la nature des données à représenter et les buts en terme d'interprétation de l'analyste ;
- établir la capacité de diverses représentations visuelles à communiquer de l'information sur les données ;
- choisir une ou plusieurs représentation(s) des données en mettant en correspondance leur capacité communicationnelle et les buts de l'analyste.

Son cadre de référence est la représentation des données au moyen de scènes naturelles en 3 dimensions. Ses exemples sont issus des sciences de l'environnement.

Robertson propose donc une **classification des types de données** et une **classification des buts de la représentation**.

Au niveau des **types de données**, il différencie tout d'abord les données *ordinales* et *nominales*

- Les données *ordinales* véhiculent une relation intrinsèque de progressivité entre des valeurs successives. Par ailleurs, elles peuvent être continues ou discrètes.
- Les données *nominales* ne sont pas associées à un ordre particulier mais représentent plutôt une classification (nominale). Elles peuvent être uni-valuées (*single-valued*) ou multi-valuées (*multi-valued*).

Robertson base sa typologie des **buts de la représentation** pour l'analyste sur la distinction entre la valeur scalaire d'une variable et sa distribution à différentes échelles. La personne qui analyse les données peut s'intéresser à différents niveaux :

- à la valeur d'une variable en un *point* ;
- à la *distribution locale* des valeurs (gradients, caractéristiques) ;
- ou à la *distribution globale* des valeurs (tendances, structure).

De plus, l'analyste peut s'intéresser à une seule variable ou à la comparaison de plusieurs variables, à des niveaux éventuellement distincts.

En ce qui concerne les propriétés des représentations graphiques, Robertson se limite dans son étude à la représentation de variables scalaires définies sur un espace à 2 dimensions. De plus, il se focalise sur la technique de représentation par scènes naturelles. Dans ce cadre, il identifie des propriétés génériques des représentations : hauteur de la surface représentée (variable ordinale, continue), matériau de couverture de la surface (variable nominale, multi-valuée) ...

A titre de discussion, nous attirons l'attention sur le fait que l'approche méthodologique de Robertson est logique et générique. Elle pourrait être utilisée comme cadre général d'une démarche de classification.

Cependant, la typologie des données proposée est relativement sommaire, bien moins élaborée que celle de Roth et Mattis par exemple. Un élément intéressant est l'introduction du concept de localité de l'analyse. Il faut également noter que l'étude des techniques de représentation est limitée à un cas très spécifique et ne peut être considérée comme une base de taxonomie générale.

L'opération de mise en adéquation d'un type de donnée et d'une représentation graphique est assez simple. Une propriété de la représentation est adéquate pour représenter une donnée si elle présente les mêmes caractéristiques selon les axes (ordinal – nominal) et (continue – discrète) que celle-ci.

Robertson mentionne lui-même la difficulté qui peut exister à identifier les buts d'une représentation. *'La visualisation est souvent opportuniste ; c.-à-d. que la personne qui interprète n'a pas toujours une bonne idée de quels attributs précis des données méritent le plus grand intérêt et dès lors elle peut souvent spécifier des buts contradictoires.'* Il propose une piste de solution de ce problème qui consisterait à baser l'évaluation de l'importance des attributs sur la nature des variables exprimée selon les deux axes précités.

2.2.4. Wehrend & Lewis (1990)

La motivation des travaux de Wehrend et Lewis (1990) est d'offrir un moyen d'identifier les techniques de visualisation pertinentes pour un problème donné. Dans cette optique, ils visent à proposer une classification des techniques de visualisation qui soit indépendante de domaines d'application particuliers. Ils supposent l'existence de problèmes génériques dans le domaine de la visualisation d'information. Ils ont pour but de proposer une taxonomie destinée à être utilisée par un être humain et pas une taxonomie qui peut servir de structure à un programme informatique.

Wehrend et Lewis mettent en parallèle le problème de la visualisation avec celui de la théorie de la mesure : *'Un système de mesure consiste en un domaine cible d'objets physiques ou conceptuels, des relations entre ces objets, un domaine de mesure d'objets mathématiques,*

des relations entre ces objets, et des correspondances qui connectent les objets et les relations dans les deux domaines. Le système doit satisfaire la contrainte mathématique que les relations dans le domaine de mesure soient cohérentes avec celles du domaine cible, de manière que (par exemple) deux objets satisfassent une relation R dans le domaine cible si et seulement si les objets correspondants dans le domaine de mesure satisfont la relation mathématique qui correspond à R . Pour exprimer ce parallèle, ils remplacent les termes ‘système de mesure’ et ‘domaine de mesure’ par les expressions ‘système de représentation’ et ‘domaine de représentation’.

Wehrend et Lewis basent leur approche sur deux classifications. La première concerne les **objets** et la seconde fait référence aux **opérations**.

La classification des **objets** fait référence à la nature des choses du domaine cible qui doivent être représentées.

- **Scalaire** (*scalar*)
- **Champ scalaire** (*scalar field*)
- **Nominal** (*nominal*)
- **Direction** (*direction*)
- **Champ de directions** (*direction field*)
- **Forme** (*shape*)
- **Position** (*position*)
- **Région ou objet ayant une étendue spatiale** (*spatially extended region or object*)
- **Structure** (*structure*)

La classification des **opérations** repose sur l'action de distinguer des problèmes pour lesquels le but de l'utilisateur (quand il regarde la représentation) diffère.

- **Identifier** (*identify*)
- **Localiser** (*locate*)
- **Distinguer** (*distinguish*)
- **Catégoriser** (*categorize*)
- **Regrouper** (*cluster*)
- **Distribuer** (*distribution*)
- **Classer** (*rank*)
- **Comparer** (*compare*)
- **Relations entre et au sein de** (*within and between relations*)
- **Associer** (*associate*)
- **Corréler** (*correlate*)

Wehrend et Lewis indiquent que le point à noter en ce qui concerne la classification des objets est que ‘des représentations différentes seront appropriées pour des objets tombant dans des classes différentes’.

Ils construisent une matrice à deux dimensions selon les deux axes de classification présentés. Ils positionnent les problèmes à résoudre dans des cellules de la matrice, de même que les techniques de visualisation. Ainsi, quand un utilisateur doit résoudre un certain type de problème, il le cherche dans la matrice et note les techniques de visualisation associées.

Ces chercheurs mentionnent néanmoins plusieurs limitations à leur approche.

- Les classifications proposées reflètent un effort pour capturer les différences superficielles entre les problèmes, plutôt que de fournir une analyse mathématique plus propre mais plus abstraite.
- Certaines distinctions faites dans la classification ne sont pas immédiatement intuitives.

- Le temps est traité comme une simple variable scalaire. Ses spécificités ne sont pas mises en évidence.
- Le traitement d'objets complexes et les opérations y afférentes est assez faible.
- Le problème de la combinaison de méthodes de représentation associées à des sous-problèmes, au sein d'une représentation globale d'une tâche n'est pas traité en profondeur.

2.2.5. Tufte (1992)

Le But de Tufte (1992, 2001) est de proposer des méthodes de conception de graphiques statistiques (*design of statistical graphics*). Pour illustrer son propos, il identifie quatre types de graphiques fondamentaux :

- **Cartes (*data maps*)**
Représentation de données géographiques.
- **Séries temporelles (*time-series*)**
Représentation de l'écoulement du temps selon un axe.
- **Représentations narratives spatio-temporelles (*space-time narrative designs*)**
Visualisation dans une seule représentation graphique de dimensions spatiales et de l'écoulement du temps.
- **Graphiques relationnels (*relational graphics*)**
Visualisation de données abstraites qui ne sont pas basées sur des analogies géographiques.

Tufte ne vise pas à proposer une taxonomie des visualisations à proprement parler. Il recense plutôt un nombre impressionnant d'exemples de visualisations qui sous-tendent sa théorie de la conception de graphiques.

2.2.6. Gruia-Catalin & Cox (1992)

Gruia-Catalin et Cox (1992) situent leur réflexion dans le domaine de la visualisation de programmes informatiques (*program visualization*). Ils définissent la visualisation comme '*une correspondance (mapping) entre des programmes et des représentations graphiques*'.

Sur base de cette définition, ils détaillent les caractéristiques définissant un système de visualisation :

- la signification accordée au concept de 'programme' ;
- le vocabulaire graphique disponible ;
- les types de correspondances possibles ;
- les moyens de construire ces correspondances.

Gruia-Catalin et Cox identifient 4 axes de classification des visualisations de programmes :

- la **portée** (*scope*) : quel aspect du programme est visualisé ?
- l'**abstraction** (*abstraction*) : quel type d'information est communiquée par la visualisation ?
- la **méthode de spécification** (*specification method*) : comment la visualisation est-elle construite ?
- la **technique** (*technique*) : comment la représentation graphique est-elle utilisée pour communiquer de l'information ?

Au niveau de la **portée**, Gruia-Catalin et Cox identifient 4 aspects d'un programme qui peuvent s'avérer intéressants à visualiser : le code, l'état des données (*data state*), l'état de contrôle (*control state*) et le comportement du programme.

Les systèmes de visualisation sont également distingués selon le **niveau d'abstraction** qu'ils supportent. Gruia-Catalin et Cox identifient 5 niveaux, tout en admettant que les frontières entre les niveaux sont imprécises.

- **Représentation directe**
Très peu de mécanismes d'abstraction sont employés, l'information originale¹³ peut souvent être facilement reconstruite à partir de la représentation graphique.
- **Représentation structurale**
Cacher ou encapsuler certains détails du programme à visualiser et utiliser une représentation directe de l'information restante.
- **Représentation synthétisée**
L'information intéressante n'est pas directement présente dans le programme mais peut en être dérivée. On représente donc le résultat d'une opération sur l'information originale.
- **Représentation analytique**
Il s'agit de capturer les propriétés hautement abstraites du programme.
- **Représentation explicative**
La représentation utilise des techniques visuelles pour illustrer le comportement du programme. Le but est que la représentation graphique '*raconte mieux l'histoire*'.

Les systèmes de visualisation diffèrent aussi selon la manière dont ils permettent la **spécification** des visualisations.

- **Prédéfinition**
Le système de visualisation est prédéfini pour représenter certaines informations sous une certaine forme. Le regardant n'a quasiment pas de contrôle.
- **Annotation**
Le système de visualisation permet au regardant d'attacher à des événements du programme des appels à des procédures permettant de construire et de modifier des images.
- **Association**
Le système de visualisation permet au regardant de définir une relation 'un à un' entre l'état du programme et les attributs d'objets graphiques.
- **Déclaration**
Le système de visualisation permet au regardant de définir des relations arbitrairement complexes entre l'état du programme et les attributs d'objets graphiques
- **Manipulation**
Le système spécifie les visualisations via l'utilisation d'exemples.

L'axe de classification associé à la **technique** précise la manière dont la représentation graphique est utilisée pour communiquer de l'information.

- **Sélection d'exemples**
Le système permet de visualiser des exemples types de fonctionnement du programme.
- **Structure de l'image finale**
Cette catégorie inclut notamment le placement sur l'écran et les caractéristiques internes du modèle graphique (dimension, point de vue).

¹³ Pour les auteurs, l'information originale est l'aspect du programme informatique que l'on doit représenter.

- **Encodage de l'information**

La classification de Tufte est reprise. Elle distingue trois moyens d'expression visuelle :

- les objets visuels et leurs attributs ;
- les relations visuelles entre ces objets ;
- les événements visuels.

- **Améliorations de la présentation**

Cette catégorie inclut les effets spéciaux utilisés pour améliorer la qualité visuelle de la présentation (ex. surfaces réfléchissantes, mise en évidence par changement de couleur...)

En guise de commentaires, nous pouvons mettre en évidence différents éléments.

Tout d'abord, si on la replace dans un contexte plus large, la portée de la visualisation fait donc référence aux données à visualiser, instanciées sur le cas particulier d'un programme informatique.

Ensuite, la réflexion sur les niveaux d'abstraction est intéressante et si on l'élargit en dehors du cas particulier de la visualisation de programmes, elle garde un réel attrait.

Enfin, l'axe de classification basé sur la 'technique' est le moins convaincant. La distinction entre les catégories ainsi que la nature même de ces catégories est discutable.

2.2.7. Lohse, Biolsi, Walker & Rueter (1990 & 1994)

a. Travaux de 1990

Dans leurs travaux de 1990, le but de Lohse et de ses collègues (1990) était de classer les représentations visuelles dans des groupes (*clusters*) homogènes. En d'autres termes, ils cherchaient à déterminer le type de connaissance communiqué par différentes représentations graphiques.

Leur travail est basé sur des expériences avec des sujets confrontés à des représentations visuelles. Il visait à '*classifier les caractéristiques de l'information visuelle que les gens utilisent pour interpréter les représentations visuelles*'. Ils ont posé comme hypothèse de travail que '*les gens stockent la connaissance visuelle de manière hiérarchique*'.

Ils ont également énoncé trois défis méthodologiques posés lors d'une classification de représentations visuelles basée sur une expérience.

- '*L'échantillon des représentations visuelles utilisé pour créer la taxonomie doit être un sous-ensemble représentatif de toutes les représentations visuelles du domaine de la taxonomie.*'
- '*L'échantillon de personnes utilisé pour créer la taxonomie doit être un sous-ensemble représentatif de la population globale des utilisateurs de graphiques.*'
- '*Des techniques différentes de collecte et d'analyse des données peuvent produire des classifications différentes pour les mêmes données.*'

Le travail expérimental de 1994 (Lohse et al., 1994) complète l'étude décrite en 1990. Sur base de la classification de 1990, ces chercheurs ont essayé d'identifier les caractéristiques qui différencient les types de représentations graphiques.

Au début de leur article de 1994, Lohse et al. proposent une classification des classifications de visualisations. C'est une réflexion très intéressante dans notre contexte car elle constitue la seule approche de méta-taxonomie que nous connaissons dans le domaine de la visualisation d'information.

Selon eux, les taxonomies de graphiques et d'images peuvent être regroupées en deux types :

- **Taxonomies fonctionnelles**
Elles se focalisent sur l'utilisation prévue et sur le but de l'objet graphique. Elles ne reflètent pas la structure physique des images et ne visent pas à correspondre à une représentation sous-jacente dans la mémoire humaine.
- **Taxonomies structurales**
Elles se focalisent sur la forme de l'image plutôt que sur son contenu.

Cette dichotomie pourrait être mise en relation avec les travaux de Baudel (2002a, 2002b) sur la représentation canonique de techniques de visualisation. Il propose un modèle de description de techniques de visualisation dont on peut déduire (avec certaines réserves) les éléments qu'elles permettent de mettre en relief. Cette réflexion essaye donc de bâtir un pont entre les approches structurales et fonctionnelles.

Notons que la taxonomie que Lohse et al. proposent en 1994 est une taxonomie structurale.

En 1990, les expériences réalisées par ces chercheurs ont conduit à identifier cinq clusters d'éléments graphiques :

- **Graphes et tables**
exemples : camembert, Gantt, matrice...
- **Cartes**
Expriment des données spatiales.
- **Diagrammes**
Décrivent la structure d'objets physiques ou des interrelations et des processus associés à des objets physiques. Ils peuvent être statiques ou dynamiques.
- **Icônes**
Donnent une interprétation ou une signification unique pour chaque image.
- **Réseaux**
Montrent les interrelations entre les composants. Des symboles indiquent la présence ou l'absence de données. Les correspondances entre les données sont indiquées par des lignes, des flèches ou par inclusion (containment).

En 1994, Lohse et al. ont identifié, sur base expérimentale, dix propriétés des représentations visuelles. Cette liste de propriétés est à la base de la détermination des catégories de représentations.

- Spatial – Non spatial
- Non temporel – Temporel
- Difficile à comprendre – Facile à comprendre
- Concret – Abstrait
- Continu – Discret
- Attractif – Non attractif
- Accentue l'ensemble – Accentue les parties
- Non numérique – Numérique
- Structure statique – Processus dynamique
- Communique beaucoup d'information – Communique peu d'information

L'étude expérimentale de 1994 a conduit à définir onze catégories de représentations visuelles :

- **Graphes**
Encodent de l'information quantitative en utilisant la position et la taille d'objets géométriques. Des données numériques à 1-D, 2-D ou 3-D sont placées selon un système de coordonnées cartésiennes ou polaires.

- **Tables**
Arrangements de mots, de nombres, de signes ou d'une combinaison de ces éléments pour montrer un ensemble de faits ou de relations dans un format compact. Les tables utilisent moins de notation symbolique que les graphes.
- **Tables graphiques**
Sous-ensemble des tables dans lesquelles des éléments graphiques (ex. ombre) sont utilisés pour représenter des données numériques.
- **Graphiques temporels**
Se différencient des tables par leur accent sur les données temporelles.
- **Réseaux**
Montrent les relations entre les composants. Des symboles indiquent la présence ou l'absence de composants. Les correspondances entre les composants sont représentées par des lignes, des flèches, la proximité, la similarité ou l'inclusion. Le système de coordonnées planaires n'a généralement pas de sémantique associée.
- **Diagrammes de structure**
Description statique d'un objet physique. Les données spatiales expriment les vraies coordonnées dimensionnelles de l'objet.
- **Diagrammes de processus**
Décrivent les relations et les processus associés à des objets physiques. Les données spatiales expriment des relations dynamiques, continues ou temporelles entre les objets.
- **Cartes**
Représentations symboliques de la géographie physique. Les cartes dépeignent les positions géographiques de caractéristiques particulières en utilisant des symboles ou des caractères.
- **Cartogrammes**
Cartes qui montrent des données quantitatives.
- **Icônes**
Attribuent une signification unique à une image. Chaque icône fournit une étiquette unique pour une représentation visuelle.
- **Images photo-réalistes**
Images réalistes d'un objet ou d'une scène.

Dans l'article de 1990, les auteurs mentionnent un parallèle avec les travaux de Bertin. Celui-ci s'intéressait uniquement aux représentations statiques, bidimensionnelles et monosémiques, ce qui exclut les icônes et les diagrammes. Lohse et al. indiquent que les trois clusters restant correspondent bien aux groupes de représentations graphiques décrits par Bertin.

Lohse et al. ont appliqué une technique de '*multi-dimensional scaling*' sur les données expérimentales. Ils ont obtenu une répartition des clusters selon deux axes principaux.

- Ils ont interprété le premier axe comme un continuum 'représentations discrètes – représentations continues'. Les graphes et les tables apparaissent sur l'extrémité 'discrète' de l'axe et les cartes sur son extrémité 'continue'. Les diagrammes se positionnent également comme des représentations continues. Les réseaux et les icônes occupent une position intermédiaire.
- Le second axe différencie principalement les icônes et les réseaux. Les graphes et tables, les cartes et les diagrammes occupent une position intermédiaire à ce niveau. Lohse et al. ont interprété cet axe comme une indication de l'effort cognitif requis pour extraire de l'information de la représentation visuelle. La signification d'une icône étant plus immédiate que celle d'un réseau. Ces auteurs concèdent cependant que la quantité d'effort cognitif requis dépend de la tâche à accomplir. Lire une la

valeur d'un point dans une table est plus facile que dans un graphe alors que déceler une tendance est plus facile dans un graphe que dans une table.

Lohse et al. discutent aussi de la relation entre la réalité et la représentation graphique (ou ses composants). Ils s'intéressent notamment à la cardinalité de cette relation. Selon eux, les représentations analogiques (ou continues) ont une correspondance de type 1-1.

En 1994, la recherche de Lohse et al. se focalise sur '*la manière dont les personnes classifient les représentations visuelles en catégories significatives et structurées hiérarchiquement*'.

Ils mentionnent cependant quatre faiblesses à leur classification.

- La taille de l'échantillon de représentations visuelles (60 éléments) influence la capacité de la classification à être généralisée.
- L'échantillon des personnes (40 individus) dont les jugements furent utilisés pour développer la classification est faible pour être représentatif de toute la gamme des utilisateurs potentiels.
- La technique de classification utilisée pour collecter et analyser les données influe la taxonomie produite.
- L'étude s'est focalisée sur la similarité perçue entre les représentations. Elle n'a pas investigué si la classification s'applique à l'interprétation des représentations visuelles.

2.2.8. Shneiderman (1996)

La proposition de Shneiderman (1996) est assez générique. Elle ne s'auto-limite pas à un sous-domaine particulier des représentations graphiques.

Schneiderman identifie deux axes de classification des visualisations d'information :

- les **types de données** à visualiser (*Data Types*) ;
- les **tâches à réaliser** par l'utilisateur (*Tasks*) ;

Au niveau des **types de données**, Shneiderman prend comme hypothèse que des utilisateurs regardent des collections d'éléments qui ont de multiples attributs. Il identifie sept types de données :

- **Données unidimensionnelles ou linéaires (1-D)**
Exemples : texte, liste de noms
- **Données bidimensionnelles (2-D)**
Exemples : cartes, plans
- **Données tridimensionnelles (3-D)**
Exemples : bâtiment, molécule
- **Données temporelles**
Elles se différencient des données 1-D par le fait que les éléments ont un commencement et une fin et que les éléments peuvent se chevaucher.
Exemples : données de gestion de projet, dossier médical
- **Données multidimensionnelles**
Les éléments avec n attributs deviennent des points dans un espace à n dimensions.
Exemple : données statistiques
- **Arbres**
Chaque élément, sauf la racine, possède un lien vers un seul élément parent. Les éléments et les liens peuvent avoir de multiples attributs.
Exemples : répertoires d'ordinateur, données de ventes
- **Réseau**
Les éléments peuvent être reliés à un nombre arbitraire d'autres éléments.
Exemple : données du web

Scheiderman précise néanmoins que '*les sept types de données sont une abstraction de la réalité et qu'il y a beaucoup de variations sur ces thèmes*'.

Schneiderman identifie sept **tâches** fondamentales dans le domaine de la visualisation d'information :

- ***Vue d'ensemble (Overview)***
Donner une vue d'ensemble sur la totalité de la collection d'éléments.
- ***Zoom (Zoom)***
Zoom sur les éléments intéressants.
- ***Filtrage (Filter)***
Filtrer les éléments inintéressants
- ***Détails à la demande (Details-on-demand)***
Sélectionner un élément ou d'un groupe d'éléments et offrir, à la demande, les détails associés.
- ***Mise en relation (Relate)***
Montrer les relations entre les éléments.
- ***Historisation (History)***
Conserver un historique des actions de l'utilisateur, pour supporter les fonctions '*undo*', '*replay*' et le raffinement successif.
- ***Extraction (Extract)***
Permettre l'extraction de sous-collections d'éléments et l'extraction des paramètres de la recherche (*query parameters*).

Nous commenterons un élément particulier de cette taxonomie : la tâche '*mise en relation*' car elle peut être mal comprise. Il s'agit en fait de la possibilité offerte par la visualisation de sélectionner un élément ou une valeur d'un attribut et d'afficher les éléments ou attributs en relation avec la sélection. Par exemple, dans la visualisation '*LifeLines*' de C. Plaisant et al. (1996), cela fait référence à la possibilité de cliquer sur un médicament et de voir les prescriptions ou les tests de laboratoire associés. Donc, cette tâche ne semble pas viser la représentation de la relation elle-même mais plutôt la représentation des éléments qui sont en correspondance via la relation.

2.2.9. Card & Mackinlay (1997)

Le but de Card et Mackinlay (1997) est de structurer l'espace de conception des visualisations d'information. Leur travail se base sur des propositions de taxonomie antérieures.

Card et Mackinlay distinguent trois **types de données** :

- ***Données nominales (N)***
Une donnée ne peuvent être qu'égal ou différente d'une autre.
- ***Données ordinales (O)***
Les données obéissent à une relation d'ordre.
- ***Données quantitatives (Q)***
Les données peuvent être manipulées arithmétiquement.
 - Variables spatiales intrinsèques (QX)
 - Coordonnées géophysiques (QXlon)
 - Autres données quantitatives
- ***NxN***
Ensemble de données nominales mis en correspondance avec lui même.
Exemple : graphes

Card et Mackinlay distinguent ensuite les données selon qu'elles constituent un ensemble de données original D ou qu'il s'agit d'un ensemble D' sélectionné à partir de D et/ou transformé par une fonction F.

$$D \rightarrow F \rightarrow D'$$

Cinq possibilités sont mentionnées pour la fonction F :

- Filtre non spécifié (f)
- Tri (fs)
- *Multidimensional scaling* (mds)
- Réduction des données via *sliders* et menus (>)
- *Slider* (sl)

Ensuite, Card et Mackinlay identifient trois grands **composants à une visualisation** :

- **Signes** (*Marks*)
 - Point (P)
 - Ligne (L)
 - Région (A) (*area*)
 - Surface (S) (*surface*)
 - Volume (V)
- **Propriétés graphiques traitées automatiquement** (*Automaticity Processed Graphical Properties*)
 - Position(X, Y, Z, T)
Utilisation non sémantique de l'espace-temps (*)
 - Encodage rétinien R (couleur, taille, forme, niveau de gris, orientation, texture)
 - Connexion (---)
 - Délimitation (*enclosure*) ([])
- **Propriétés graphiques dont le traitement est contrôlé** (*Controlled Processing Graphical Properties*)
Exemple : texte

La taxonomie proposée par Card et Mackinlay présente l'avantage d'offrir un formalisme qui permet d'identifier clairement la position d'une visualisation dans l'espace de conception (cf. Table 7).

Le formalisme précise sous la forme d'un tableau :

- le type de données,
- le type de transformation des données avant la visualisation,
- la correspondance entre les données et les composants de la visualisation.

Card et Mackinlay utilisent le formalisme pour décrire tout un ensemble de visualisations proposées dans la littérature.

Table 7: Exemple d'utilisation du formalisme de Card & Mackinlay

<i>Données</i>				<i>Propriétés automatiques</i>						<i>Contrôlées</i>	
Nom	D	F	D'	X	Y	Z	T	R	---	[]	CP
Ozone	Q		O					C			
QXlon	QXlon	f	QXlon	P							
Année	Q	>	Q	P							
Set	NxN	f	xXy	*	*				L		
Fichiers	NxN	tree	xXy	*	*					*	
...											

Card et Mackinlay mentionnent que deux niveaux d'analyse n'ont pas été pris en compte dans leur article :

- la structure organisationnelle plus large des espaces d'information ;
(*exemples : espace d'information, espace de travail, documents...*)
- l'organisation des tâches des utilisateurs.

De plus, ils précisent qu'ils ont donné leur préférence à une notation utilisable, c.-à-d. qui offre un réel pouvoir d'expression sans tomber dans une complexité trop pénalisante. De ce fait, certains éléments de différenciation peuvent avoir été négligés. Ils se sont focalisés sur la mise en correspondance des données avec leur présentation.

2.2.10. Tweedie (1997)

Tweedie (1997) avait pour but de caractériser les visualisations interactives.

Elle étudie trois aspects en particulier : les données utilisées pour créer la représentation, les formes d'interactivité disponibles pour l'utilisateur, l'information d'entrée/sortie qui est explicitement représentée et les questions possibles sur les données.

Au niveau de la **représentation des données**, elle distingue tout d'abord la représentation des **valeurs des données** et celles de la **structure des données**.

Les premières montrent les relations entre des sous-ensembles des données. De telles représentations tendent à mettre en correspondance des attributs à des axes. (Exemples : Scatterplot, histogramme, coordonnées parallèles...)

Les secondes représentent les relations au sein d'un seul ensemble. A ce niveau, Tweedie reprend la taxonomie de Bertin :

- Représentation rectiligne
- Représentation circulaire
- Représentation en motifs ordonnés (*ordered pattern*)
Exemple : arbre
- Représentation en motifs non ordonnés (*unordered pattern*)
Exemples : réseaux, diagramme de Venn
- Représentation en stéréogramme
Représentations où la structure suggère un volume

Dans certains cas, les utilisateurs sont intéressés par des résumés des données sous la forme de statistiques. Tweedie parle alors de **représentation de données construites**. De manière similaire, il peut être parfois utile de combiner de l'information à propos de la manière dont une structure est en relation avec une autre.

Tweedie mentionne également le cas de la **représentation de données converties**.

A ce moment, on dispose d'un ensemble de données sans connaître les relations structurales qui y sont présentes. Il peut cependant être instructif de représenter la structure (*derived structure*) de l'ensemble, calculée à partir des données. (Exemple : structure d'un ensemble de documents).

Dans d'autres cas, on dispose d'une bonne compréhension des relations structurales du problème et leur compréhension peut être facilitée par une représentation de certaines instanciations de cette structure (*derived values*). (Exemple : représentation d'instanciations d'équations algébriques.)

Le second grand axe de classification proposé par Tweedie fait référence à **l'interactivité**.

Elle explique que les outils interactifs fournis pour agir sur une visualisation peuvent être décrits en termes d'équilibre de contrôle (*balance of control*) entre l'utilisateur et l'ordinateur. Elle reprend la proposition de Lunzer pour décrire les formes d'interaction :

- **Interaction manuelle** (*manual*)
exemple : faire glisser quelque chose sur l'écran avec la souris
- **Interaction mécanisée** (*mechanized*)
exemple : faire une sélection avec un outil comme un curseur (slider)
- **Interaction enseignable** (*instructable*)
exemple : formules dans une feuille de calcul
- **Interaction guidable** (*steerable*)
exemple: utiliser un algorithme défini pour s'exécuter d'une certaine manière
- **Interaction automatique** (*automatic*)
exemple : utiliser un algorithme qui s'exécute automatiquement

Par ailleurs, Tweedie précise que la plupart de l'interactivité proposée dans les outils de visualisation fait usage de 'manipulation directe' (*direct manipulation*), c.-à-d. d'une réplification littérale du comportement physique dans le monde réel. Elle évoque ensuite le fait que des techniques plus récentes permettent de la 'manipulation indirecte' (*indirect manipulation*), qui ne repose pas sur des métaphores physiques directes (exemple : déformation d'objets dans des environnements virtuels).

Pour le troisième et dernier axe de classification, Tweedie reprend les travaux généraux de Draper sur les **entrées/sorties** et les applique pour classifier les différents types de feedback visuels possibles offerts par une représentation :

- **Entrée → Entrée**
Exemple: mettre en relation deux entrées utilisant la main (handed input).
- **Entrée → Sortie**
Exemple: mettre en relation un curseur et un histogramme.
- **Sortie → Entrée**
Exemple: relier un message d'erreur avec sa cause.
- **Sortie → Sortie**
Exemple: Relier deux affichages de sortie (output displays).

Tweedie insiste également sur l'importance de permettre à l'utilisateur d'explorer l'historique des relations 'entrées-sorties' qu'il a expérimentées avec la représentation. Dans le même contexte, elle mentionne l'intérêt à offrir à l'utilisateur un moyen de prendre conscience de l'espace des données qu'il n'a pas encore parcouru.

En ce qui concerne les **questions possibles sur les données**, Tweedie reprend la classification de Bertin selon deux axes.

Selon le premier axe, les questions peuvent être posées à trois niveaux différents :

- *à propos d'un seul élément,*
- *à propos d'un sous-ensemble d'éléments,*
- *à propos de tous les éléments.*

Selon le second axe, on distingue :

- *les questions qui portent sur les objets,*
- *et les questions qui portent sur les attributs des objets.*

En combinant les deux axes, on obtient six cas possibles.

Dans ses discussions, Tweedie précise que Bertin avait négligé dans sa taxonomie le fait qu'il y ait non seulement des données brutes (*raw data*) et des données structurales (*structural data*) mais aussi des méta-données issues de la transformation des données brutes. Elle introduit les concepts de données construites (*constructed data*), qui sont des transformations qui n'altèrent pas la forme des données (valeur → valeur dérivée, structure → structure dérivée) et de données converties (*converted data*), qui sont des transformations qui modifient la forme des données (données → structure dérivée, structure → données dérivées).

2.2.11. Chi (2000)

Le but de Chi (2000) était de proposer une taxonomie avec une portée opérationnelle. Son travail se base sur des propositions de taxonomie antérieures.

Sa taxonomie repose sur le modèle d'état des données (*Data State Model - DSM*). Le modèle DSM différencie chaque technique de visualisation selon :

- **les phases des données** (*Data Stages*)
 - **Valeur** (*value*)
Données brutes
Exemple : système de fichiers
 - **Abstraction analytique** (*analytical abstraction*)
Données à propos des données, métadonnées
Exemple : graphe
 - **Abstraction de visualisation** (*visualization abstraction*)
Information qui est visualisable à l'écran en utilisant une technique de visualisation.
Exemple : Hiérarchie en arbre
 - **Vue** (*view*)
Produit final de la visualisation, où l'utilisateur voit et interprète l'image qui lui est présentée.
Exemple : Treemap, arbre hyperbolique...
- **les opérateurs de transformation des données** (*Data Transformation Operators*)
 - **Transformation des données**
Génère une forme d'abstraction analytique à partir de la valeur (souvent par extraction)
 - **Transformation de la visualisation**
Prend une abstraction analytique et la réduit en une forme d'abstraction de visualisation.
 - **Transformation de la cartographie visuelle** (*visual mapping*)
Prend de l'information dans un format visualisable et présente une vue graphique.

- **les opérateurs internes aux phases** (*Within Stage Operators*)

Ils ne changent pas la structure des données.

- **Opérateurs internes à la valeur** (*Within Value*)
Exemple : filtrage des valeurs
- **Opérateurs internes à l'abstraction analytique** (*Within Analytical Abstraction*)
Exemples : normalisation de l'échantillon, filtrage dynamique des valeurs, regroupement de nœuds...
- **Opérateurs internes à l'abstraction de visualisation** (*Within Visualization Abstraction*)
Exemple : filtrage dynamique des valeurs...
- **Opérateurs internes à la vue** (*Within View*)
Exemples : browse, scroll, rotate, dynamic view-filtering...

Chi explique que les taxonomies antérieures à la sienne permettaient surtout à celui qui implémente d'identifier les techniques de visualisation applicables au domaine de données qui le concerne. Par contre, elles n'aidaient pas à comprendre comment appliquer et implémenter ces techniques.

Chi explique que sa taxonomie est la plus détaillée et la plus complète jamais proposée jusqu'à ce moment (c.-à-d. en 2000). Il argumente en précisant qu'elle est la plus détaillée dans le sens où il a différencié chaque technique non seulement par type de données mais aussi par étapes de traitement opérationnelles (*processing operating steps*). Selon lui, sa taxonomie est aussi la plus complète car elle catégorise les techniques bien connues présentées dans les travaux antérieurs.

Le modèle '*Data State Model*' a été utilisé pour étudier 36 techniques de visualisation. Il a donné lieu à l'application '*Visualization Spreadsheet*'.

La taxonomie de Chi est intéressante dans le sens où elle ajoute les étapes opératoires pour mettre en œuvre les techniques de visualisation. Chi montre que certaines techniques partagent certaines étapes opératoires qui pourraient être standardisées dans un système. Il indique aussi que si un système de visualisation est conçu en modules organisés selon sa taxonomie, il permet une grande réutilisabilité.

2.2.12. Grinstein et al. (2001 & 2002)

Grinstein et ses collègues (Grinstein et al., 2001 ; Hoffman et Grinstein, 2002) s'intéressent aux techniques de visualisation à nombre élevé de dimensions (*high-dimensional visualization techniques*) et plus particulièrement aux '*visualisations qui peuvent présenter un grand nombre de dimensions ou de paramètres des données*'. Ils précisent également qu'ils se focalisent sur les tables de données numériques bien que beaucoup de techniques parmi celles qu'ils citent s'appliquent aussi aux données catégorielles (*categorical data*).

Le but des travaux présentés en 2001 par Grinstein, Trutschl et Cvek, était de définir la dimensionalité des techniques de visualisation. Dans ce contexte, ils insistent sur la nécessité de distinguer deux concepts : espace de définition des données à nombre élevé de dimensions (*high-dimensional data*) et visualisation à nombre élevé de dimensions (*high-dimensional visualization*). Plus précisément, ils proposent la définition suivante. '*Soit un espace à n dimensions, la dimension intrinsèque d'une visualisation est définie comme le plus grand k , $k \leq n$, pour lequel un ensemble de k vecteurs unitaires (dans cet espace à n dimensions) peut être identifié (perçu) de manière unique dans la visualisation.*'

Les articles de 2002, rédigés par Hoffmann et Grinstein et par Grinstein et Ward appartiennent à un ouvrage collectif sur la visualisation d'information dans le domaine du *Data Mining* et du *Knowledge Discovery*.

Dans leur article de 2002, Grinstein et Ward, proposent une classification des techniques de visualisation selon trois axes : la **focalisation**, la **dimension du stimulus** et l'**importance de l'interaction**.

Au niveau de la **focalisation**, ils distinguent :

- **le focus géométrique (Geometric focus)**
Inclut les techniques de représentation des données comme des scatter plots, des lignes, des surfaces ou des volumes. Typiquement, les données ont plusieurs champs qui peuvent être mis en correspondance avec des axes pour être affichés.
- **le focus symbolique (Symbolic focus)**
Représentent surtout des données non-numériques, en utilisant des pixels, des icônes, des tableaux ou des graphes (réseaux, relations...). Le focus est plus sur l'affichage des relations entre les éléments. Les considérations topologiques peuvent être plus importantes.

En ce qui concerne la **dimension du stimulus**, ils proposent quatre valeurs :

- *les Stimulus 2D*
- *les Stimulus 3D*
- *les stimulus stéréoscopiques*
- *l'immersion*

Enfin, à propos de l'**importance de l'interaction**, ils différencient :

- *l'affichage de données statiques*
- *l'affichage de données dynamiques*

Sur base d'une revue de diverses taxonomies, Grinstein et ses collègues proposent également une classification des visualisations à haute dimension. La taxonomie présentée ci-dessous résulte de la fusion des deux taxonomies proposées dans les articles de 2001 et 2002. Nous avons fusionné ces deux classifications car elles étaient très proches l'une de l'autre.

- **2D and 3D Scatterplots (2001, 2002)**
Projection des données sur un espace à 2 ou 3 dimensions, représenté dans le format (x,y) ou (x,y,z).
- **Matrix of scatterplots (2001, 2002)**
Tableau de scatterplots affichant toutes les combinaisons deux à deux de dimensions ou de coordonnées.
- **Heat map (2001)**
Tableau de cellules où chaque cellule est colorée sur base d'une valeur de la donnée ou d'une fonction sur les données.
- **Height map (2001)**
Tableau de cellules où la valeur associée à chaque cellule est représentée par une hauteur.
- **Table Lens (2001)**
Tableur dans lequel chaque cellule peut être affichée de manière optionnelle en utilisant une ligne dont la longueur dépend de la valeur numérique de la cellule et dont la couleur peut représenter un autre attribut des données.
- **Survey plots (2001, 2002)**
Représentation qui consiste généralement en n régions rectangulaires, chacune d'entre elles représentant une dimension dans une matrice de données. Un axe est utilisé pour représenter les données dans chaque dimension. Dans chaque dimension, les points de données (cf. line graph) sont projetés sur l'axe selon une droite qui est

visualisée. Les lignes de projection ont une longueur proportionnelle à la valeur de la donnée associée au point projeté.

- **Permutation matrix (2002)**
Représentation similaire au 'survey plot', dans laquelle la hauteur des barres correspond aux valeurs des données. Les motifs dans les données apparaissent en permutant ou en triant les lignes ou les colonnes de la matrice.
- **Iconographic displays (2001, 2002)**
Représentation graphique visualisant des données à haute dimension en laissant chaque dimension d'un enregistrement conduire un paramètre ou un attribut d'une entité (pixel, icône ou glyphe) et en affichant simultanément un ensemble de ces entités à l'écran. Il en existe deux variantes :
 - Certaines dimensions de l'ensemble de données à n dimensions sont mises en correspondance avec certaines caractéristiques du glyphe ou de l'icône.
Exemple : Faces de Chernoff
 - Les glyphes ou les icônes forment un affichage dense, dont la texture représente des caractéristiques de l'ensemble de données.
- **Dimensional stacking (general logic diagrams) (2001, 2002)**
Projection 2D ou 3D des données où les dimensions sont imbriquées les unes dans les autres. La grille 2D est successivement divisée en ensembles de rectangles imbriqués, représentant des dimensions catégorielles ou des attributs des données.
Temple MVV est une extension du 'dimensional stacking' combinée avec les concepts d'histogrammes et de 'treemaps'.
Le paradigme de 'dimensional stacking' a été étendu aux environnements 3D interactifs pour donner le système N-vision.
- **Fractal Foam (2002)**
Visualisation qui représente la corrélation entre les dimensions de manière récursive. Une première dimension est représentée par un cercle coloré. Les autres dimensions sont représentées comme des cercles plus petits qui tournent en sens anti-horlogique autour du cercle principal. La taille des petits cercles représente le coefficient de corrélation entre la première dimension et chacune des autres. Le processus est répété récursivement. D'autres propriétés statistiques des données peuvent être représentées par des propriétés des cercles (ex. rugosité). La visualisation ne représente pas les données mais des propriétés statistiques de celles-ci.
- **Parallel coordinates (2001, 2002)**
Utilisation d'axes parallèles pour représenter les dimensions d'un ensemble de données multidimensionnel. Une polyligne constituée d'au plus $(n-1)$ segments connecte les axes pour représenter un point à n dimensions.
Une variante consiste à ne pas utiliser des axes parallèles pour représenter les dimensions mais les rayons d'un cercle (circular parallel coordinates).
- **Line graph, multiple line graph (2001, 2002)**
Lignes qui affichent la valeur de fonctions uni-valuées ou continues par morceaux relatives à une dimension/variable (single-valued or piecewise continuous functions of one variable). Pour représenter les différentes dimensions de l'ensemble de données, plusieurs lignes peuvent être représentées sur le même graphe.
- **Bar charts, histograms (2002)**
Un 'bar chart' est un 'line graph' dans lequel la région sous la ligne apparaît comme remplie. Un histogramme est un 'bar chart' dans lequel la valeur d'une barre représente une somme de points.
- **Pixel techniques, circle segments (2001, 2002)**
Arrangement des données dans un espace délimité, à partir de l'origine, en utilisant des techniques variées telles que des segments récursifs, en spirale ou des segments de cercle. L'interprétation de la position (x,y) de la cellule dépend de la mise en

correspondance choisie. Chaque dimension est associée à un pixel et sa valeur est associée à une échelle de couleurs.

Exemple : VisDB.

- **Multi-dimensional scaling and Sammon plots (2001)**
Projection des données sur un espace 2D ou 3D après un traitement MDS. La technique MDS conserve la distance métrique entre les points, c.-à-d. que les points proches l'un de l'autre dans l'espace à n dimensions le seront aussi dans la projection 2D ou 3D.
Sammon plot est une variante non-linéaire de MDS.
- **Polar charts (2001)**
Graphe basé sur des coordonnées polaires. Les données sont affichées sous la forme de points (éventuellement reliés par des lignes).
- **RadViz (2001, 2002)**
Technique d'affichage qui place des points d'ancrage associés aux dimensions sur le périmètre d'un cercle. Un ressort virtuel relie chaque ancre à chaque point. Les constantes des ressorts sont utilisées pour représenter les valeurs relatives des données dans chaque dimension. La position d'un point est calculée en annulant la somme des forces des ressorts qui lui sont attachés (un ressort par dimension).
- **PolyViz (2001, 2002)**
Extension de RadViz où chaque dimension est associée à une ligne d'ancrage plutôt qu'à un point (la figure globale est un polygone plutôt qu'un cercle). Des constantes de ressorts sont utilisées le long de la ligne d'ancrage, correspondant à toutes les valeurs de la dimension associée.
- **Principal component and principal curve analysis (2001)**
Technique d'analyse qui identifie un espace de dimension moindre que celui des données dans lequel la variance des données est préservée.
- **Grand Tour and Projection Pursuit (2001, 2002)**
Grand Tour est une animation de 'scatter plots' 2D (ou 3D), dans lesquels les axes x , y (et z) représentent des combinaisons linéaires des N dimensions.
Projection Pursuit produit aussi des projections de données mais dans lesquelles un but particulier conduit les projections (ex. discriminer deux classes).
- **Dendogrammes (2002)**
Diagrammes en arbre basés sur la manière dont les données individuelles se regroupent.
- **Kohonen self-organizing maps (2001)**
Combinaison d'une technique analytique (réseau de neurones) et d'une technique graphique pour regrouper les données dans le but de réduire la taille de l'ensemble initial. Elle permet la construction non supervisée puis l'affichage de groupes de données (clusters) dans un espace 2D. Les données semblables sont regroupées dans la représentation.
- **Andrews's Curves (2002)**
Les courbes d'Andrews affichent chaque point à n dimensions comme une courbe, en utilisant la fonction où le point à n dimensions est calculé selon une formule $X=(x_1, x_2, \dots, x_n)$. C'est similaire à une transformée de Fourier des point de données.
- **Virtual Reality, PITS (Populated Information Terrains) Visualizations (2002)**
Application des paradigmes de la réalité virtuelle à la visualisation d'information. Le paradigme des scènes naturelles (natural scene) est associé à cette catégorie. Il s'agit de représenter les informations en les mettant en correspondance avec des éléments dans des scènes naturelles.

Néanmoins, il faut noter que, dans l'article de 2002, un regroupement de certaines techniques de visualisation s'ajoute au simple recensement des techniques présenté dans celui de 2001.

Ce regroupement n'est cependant pas clairement énoncé et il est difficile de savoir s'il couvre tout l'ensemble des techniques de visualisation citées.

- **Table Visualizations**
Concerne les visualisations de tables 2D de données, définies par M lignes et N colonnes. Les visualisations de tables de données ne sont pas hiérarchiques. Elles ne contiennent pas explicitement de structure interne ou de liens. L'ensemble de données a une taille finie (M et N sont bornés). Les données peuvent être vues comme M points à N dimensions.
Exemples : Line graph, scatter plot
- **Recursive Table Visualizations**
L'affichage impose une structure hiérarchique des données, de sorte que certaines valeurs dimensionnelles sont imbriquées dans l'affichage d'autres dimensions. Ces visualisations sont aussi appelées 'axes hiérarchiques'.
Exemples : Dimensional Stacking, N-Vision
- **Clustering Visualizations**
Visualisations qui essayent de montrer les clusters possibles dans un ensemble de données.
Exemples : Dendogram, Kohonen self-organising map
- **Nonlinear Projection Visualizations**
Techniques qui réduisent les dimensions de l'espace des données en projetant les données en deux dimensions, tout en essayant de préserver les distances entre tous les points.
Exemple : Multi Dimensional Scaling

Les travaux de Grinstein et de ses collègues appellent à quelques commentaires.

Dans leur article de 2002, Grinstein et Ward précisent que toutes les techniques ne se retrouvent pas dans précisément dans la classification des techniques de visualisation proposée. Ils ajoutent que les lignes de séparation ne sont pas précises et que plusieurs visualisations sont des hybrides issus de plusieurs techniques.

Quelques années plus tard, en 2004, Grinstein, Keim et Ward reprennent la taxonomie des visualisations à haute dimension en précisant que certaines techniques sont des techniques de visualisation pures (*scatterplots, matrix of scatterplots, statistical charts, line and multi-line graphs, parallel coordinates, circle segment, polar chart, survey plots, heatmaps, height maps, iconographic displays, RadViz, PolyViz*), d'autres sont intégrées avec de l'analyse des données (*projection pursuit, dimensional stacking, Sammon plots, MDS, PCA and principal curves, self organising maps*) et enfin les dernières sont associées à des interactions (*selection, probing, querying, grand tours, non-linear zooms*).

2.2.13. Tory & Möller (2002)

Tory et Möller (2002) critiquent les taxonomies des techniques de visualisation proposées jusqu'alors car la terminologie qu'elles utilisent est trop vague. Ils mentionnent par exemple le manque de clarté dans la définition traditionnelle des deux grandes disciplines de la visualisation : la visualisation scientifique (*scientific visualization*) et la visualisation d'information (*information visualization*).

De plus, selon eux, '*la dépendance des schémas de classification existants vis-à-vis du type des données est problématique car les données font l'objet d'une interprétation significativement importante pendant le processus de visualisation, à la fois par les programmeurs des algorithmes de visualisation et par les utilisateurs des systèmes de visualisation*'.

Tory et Möller proposent une réflexion générale préalable à la construction d'une taxonomie des visualisations. '*En analyse visuelle de données, l'objet d'étude est l'idée ou l'objet physique investigué. Habituellement, parce qu'il ne peut pas être étudié directement, l'objet*

d'étude est analysé indirectement par le biais d'un ensemble de valeurs discrètes, les données. Les algorithmes de visualisation créent des représentations visuelles des données en utilisant un modèle de données. Les gens ont des idées préconçues à propos de l'objet d'étude et des interprétations des données qui affectent leur compréhension de ce que les données représentent. Ce modèle conceptuel de l'objet d'étude est ce qui détermine quels modèles de données sont implémentés par le programmeur de l'algorithme de visualisation et quelles techniques de visualisation sont choisies par la personne qui les regarde.'

L'idée de base de Tory et Möller est de baser la taxonomie sur les caractéristiques des modèles de données plutôt que sur les caractéristiques des données elles-mêmes. Ils justifient cette approche en expliquant : *'comme un jeu de données peut être conceptualisé de différentes manières, il est plus logique de classifier les techniques de visualisation sur base des modèles de données qu'elles utilisent, plutôt que sur les données elles-mêmes.'*

La taxonomie des algorithmes de visualisation proposée distingue tout d'abord les modèles de données selon leur continuité. Ils différencient les **modèles de données continues** et **discrètes**.

Dans le premier cas, on considère que les **données modélisées** sont **continues**. Ils considèrent alors deux axes de classification :

- **Nombre de dimensions**
 - *Nombre de variables indépendantes*
 - *Nombre de variables dépendantes*
- **Type de données**
 - *Scalaire*
 - *Vecteur*
 - *Tenseur*

Table 8: Modèle de données continues (cf. [197], p. 3)

# indep. var.	# dép. variables	1			Many
	Data type	Scalar	Vector	Tensor	Multivariate
	1D	- Line graph			Combine scalar, vector & tensor methods
	2D	-Colour map - Isolines	- LIC - Particle traces		
	3D	- Volume rendering - Isosurfaces	- Glyphs	- Tensor ellipsoids	
	nD	Multiple 1D, 2D, or 3D views			

Dans le second cas, on considère que les données modélisées sont discrètes.

Table 9: Modèle de données discrètes (cf. [197], p. 3)

Connected	Unconnected		
Graph & Tree Visualizations	# var.	2D	- Scatter plot - Bar chart
- Node-link diagrams (2D and 3D) - Space-filling mosaics - Hierarchical graphs		3D	- 3D scatter plot - 3D bar chart
		nD	- Multiple views - Aggregation - Glyphs - Dense pixel displays - Parallel coordinates

Tory et Möller font remarquer que leur schéma de classification n'est pas complet. Il devrait notamment être complété en décrivant des niveaux inférieurs de hiérarchie, en y incluant plus de techniques de visualisation, en y ajoutant les catégories de tâches...

Ils font aussi remarquer que leur taxonomie définit la dimensionnalité du modèle de données différemment pour les modèles discrets et continus. Pour les modèles continus, la dimension correspond au nombre de variables indépendantes, alors que pour les modèles discrets, elle correspond au nombre total de variables.

2.2.14. Tilley & Huang (2002)

Tilley et Huang (2002) se focalisent sur les techniques de visualisation de programmes informatiques.

Leur but est de construire une classification orientée 'tâche' (*task-oriented classification*), ce qui, selon eux, présente l'avantage de mettre en correspondance des activités habituelles associées à la compréhension des programmes (ex. 'montrer quelles variables sont utilisées par une fonction donnée') avec des types spécifiques de logiciels de visualisation. Compte tenu de la difficulté à élaborer une telle classification orientée 'tâche', Tilley et Huang ont commencé par créer une classification descriptive des techniques de visualisation de programmes.

Tilley et Huang ont choisi de partitionner les visualisations de programme sur base du niveau d'interaction entre l'utilisateur final et la documentation graphique générée. Ils distinguent trois classes de visualisations : statiques, interactives et éditables.

- **Statiques :**
A part sélectionner les objets qui apparaissent dans la visualisation, l'utilisateur n'interagit en aucune manière avec l'image statique. La représentation est seulement incluse comme documentation graphique, accessible en lecture seule.
- **Interactives :**
La documentation graphique est interactive, ce qui permet à l'utilisateur de naviguer à travers le diagramme. Il peut traverser le graphe en passant d'un nœud à un autre au travers de liens. Pendant la traversée, la structure fondamentale du graphe reste inchangée.
- **Éditables :**
Les visualisations éditables permettent à l'utilisateur de modifier le graphe généré lui-même, par exemple en ajoutant de nouveaux nœuds ou de nouveaux arcs. Si elles sont connectées à une base de données, elles peuvent être sauveées et réutilisées. Les graphes éditables sont parfois référencés comme des '*live documents*'.

Dans ce contexte, ils mentionnent que la guidance que l'utilisateur reçoit de l'outil de visualisation pendant la génération de la documentation graphique est aussi un élément important.

Tilley et Huang insistent sur le fait que '*leur classification descriptive est un moyen d'atteindre un but et non un but en soi*'. Le but final étant de répondre à la question : 'Quels types de visualisation offrent le plus de support pour les tâches de compréhension d'un programme ?' Ils mentionnent également que '*les descripteurs de la classification sont fortement influencés par les caractéristiques du domaine d'application de leur partenaire industriel*'.

Par ailleurs, Tilley et Huang admettent que les trois classes de visualisation qu'ils proposent ne sont pas strictement disjointes mais ils affirment que la plupart des outils peuvent être aisément positionnés dans une des trois.

Tilley et Huang font indirectement référence, dans la description des visualisations statiques, au problème crucial du choix du point de vue sur le problème à visualiser par la personne qui conçoit la visualisation.

2.2.15. Pfitzner, Hobbs & Powers (2003)

Pfitzner et ses collègues (2003) étudient la question de savoir quel est le meilleur moyen pour afficher de l'information pour une application donnée.

Ils justifient leur travail, qui repose sur des propositions antérieures, en mettant en évidence le fait que les taxonomies proposées jusqu'alors ne s'intéressaient qu'à un ou deux facteurs spécifiques de la conception d'interface. Leur objectif est de proposer un large cadre de référence pour le domaine tout entier de la visualisation d'information.

Leur recherche est issue de travaux dans le domaine de la visualisation pour la recherche d'information (*information retrieval*).

Pfitzner et al. identifient cinq facteurs qui doivent être considérés lorsqu'on conçoit une interface de visualisation d'information : les données, les tâches, l'interactivité, le niveau d'expertise de l'utilisateur et le contexte d'utilisation.

En premier lieu, Pfitzner et al. indiquent que les **données** possèdent deux types de caractéristiques principales :

- **Type de données** (*data type*)
 - Données de bas niveau (*low-level data*)
 - Objet
 - Attribut
 - Données de haut niveau (*high level data*)
Elles sont appelées méta-données.
- **Relation entre les données** (*data relationship*)
Elle décrit la manière dont sont structurées les collections de données
 - Linéaire (*linear*)
 - Circulaire (*circular*)
 - Arbre ordonné (*ordered tree*)
 - Arbre non ordonné (*un-ordered tree*)
 - Hyper-espace (*hyperspace*)

Pfitzner et al. s'intéressent ensuite aux **tâches**, qu'ils définissent comme '*ce que les utilisateurs veulent accomplir et comment ils l'accomplissent en utilisant tout ou partie de la fonctionnalité de l'interface*'. Ils reprennent telle quelle la taxonomie de Schneiderman (1996) pour cette composante (cf. 2.2.8, p. 52).

En troisième lieu, ils reprennent la réflexion de Tweedie (1997) et proposent de définir le degré **d'interaction** selon un continuum segmenté en cinq zones.

Puis, ils précisent que lors de la conception d'une visualisation d'information, il est indispensable de tenir compte du **niveau d'expertise de l'utilisateur** en matière de visualisations. Ils répartissent les niveaux d'expertise selon un continuum qui s'étend de 'utilisateur novice' à 'utilisateur expert'. Néanmoins, ils n'approfondissent pas cette notion et ne proposent pas de classification des utilisateurs.

Enfin, ils définissent la notion de '**contexte**' comme '*les facteurs qui sont externes à l'utilisation de l'artefact ordinateur mais qui influencent l'utilisateur dans son utilisation de la visualisation*'. Ils proposent cinq dimensions contextuelles que le concepteur de visualisations d'information doit considérer :

- **Expérience vécue** (*life experience*)
exemples : compétence informatique, expérience dans le domaine...
- **Intention** (*intent*)
Décrit ce que l'utilisateur pense qu'il veut obtenir en utilisant la visualisation.
- **Besoin** (*need*)
Décrit le besoin réel de l'utilisateur pour utiliser la visualisation.
- **Historique** (*history*)
Décrit l'utilisation des visualisations par l'utilisateur, en termes de période d'utilisation.
- **Appareil** (*device*)
Décrit le type d'appareil utilisé pour afficher la visualisation.

Après avoir discuté des facteurs dont il faut tenir compte lors de la conception d'une technique de visualisation, ces chercheurs proposent une classification de celles-ci en termes de présentation de l'information, de dimensions d'affichage, de dynamique de l'affichage et d'animation de l'affichage.

Pfitzner et al. considèrent qu'il existe deux manières générales de **présenter de l'information** à un utilisateur :

- sous forme de **texte** ou
- sous forme de **représentation picturale abstraite**.

Ils observent que la plupart des visualisations d'information combinent les deux.

Il proposent de définir un continuum relatif à la forme de la présentation (*Presentation Form Continuum*). Ce continuum s'étend entre les valeurs 'présentation textuelle' et 'présentation graphique'.

Pfitzner et al. se basent sur le travail de Bertin pour décrire les **dimensions offertes par un affichage** sur écran.

- **Variables de l'image**
 - Plan (*plane*)
Coordonnées qui identifient la position d'un composant affiché.
 - Taille (*size*)
Pourcentage de la surface de l'écran que le composant utilise.
 - Valeur (*value*)
Tout indicateur du composant qui communique une différence de valeur entre lui et d'autres composants.
 - Relation (*relationship*)
Position relative d'un composant par rapport à un autre.

- **Variables différentielles**
 - Texture (*texture*)
Motif de surface utilisé pour différencier les composants et communiquer une sémantique supplémentaire.
 - Couleur (*colour*)
Toute couleur dans la gamme visible.
 - Orientation (*orientation*)
Positionnement angulaire du composant par rapport aux frontières de l'affichage ou à d'autres composants.
 - Forme (*shape*)
Forme prise à l'écran par le composant visuel.

Pfitzner et al. distinguent deux catégories d'affichage en termes de **dynamique** :

- **Affichage statique** (*static display*)
Ne permet pas d'interaction.
- **Affichage dynamique** (*dynamic display*)
Permet de l'interaction.

Enfin, l'**animation de l'affichage** (*display animation*) fait référence à la présence de mouvement et de changement dans la visualisation.

A titre de commentaire, nous attirons l'attention sur le fait qu'il ne s'agit pas vraiment d'une taxonomie des visualisations d'information mais plutôt un assemblage de taxonomies antérieures. Certains axes concernent directement la classification des visualisations alors que d'autres sont plutôt les facteurs influençant leur conception ou leur utilisation réelle.

Pfitzner et al. discutent aussi du nombre de concepts – qu'ils appellent '*data facts*' – que l'utilisateur doit appréhender à chaque instant dans la visualisation mais ils ne s'attardent pas sur cette notion.

2.2.16. Wenzel, Bernhard & Jessen (2003)

Wenzel et ses collègues (2003) se focalisent sur les techniques de visualisation utilisées dans le contexte de la simulation en production et logistique.

Une des références de base de leur travail est constituée par les travaux de l'association VDI (*Verein Deutscher Ingenieure*) et en particulier ceux du groupe de travail 'Simulation et Visualisation'.

Ils identifient tout d'abord un ensemble de cas d'utilisations possibles de visualisations :

- outil d'acquisition (*acquisition tool*)
- utilitaire de vérification dans un processus de modélisation (*verification utility*)
- utilitaire de validation (*validation utility*)
- outil d'analyse pour expliquer des éléments du problème (*analysis tool*)
- outil de communication (*communication tool*)
- outil de présentation (*presentation tool*)
- outil de formation (*training tool*)

Ensuite, ils identifient différents critères de classification des visualisations d'un point de vue graphique.

Le premier d'entre eux est la **dimension** :

- **1-D** : présentation qui détermine seulement une dimension
- **2-D** : présentation de graphes bidimensionnels et dispositions de modèle (*model layouts*)
- **2 ½ D** : présentation 3-D développée à partir d'une présentation 2-D
- **3-D** : présentation spatiale avec le libre choix de la perspective

Le second critère est le type de **représentation** :

- **Représentation symbolique**
La signification d'un symbole est déterminée par un code.
 - Symbole de type caractère
c.-à-d. élément qui représente de l'information sur base d'une quantité finie convenue (ex. caractère alpha-numérique)
 - Symbole abstrait
c.-à-d. image 2-D, 2 ½ D ou 3 D ayant une signification spécifique dans une certaine application
- **Représentation iconique**
Une icône possède sa signification sans convention de code.
 - Icône stylisée
 - Icône proche de la réalité
- **Représentation photo-réaliste**

Le troisième critère est le **format d'affichage** :

- **Police (font)**
Série finie de caractères.
- **Table / tableur (table / spreadsheet)**
Arrangement visuel structuré de caractères.
- **Graphique (chart)**
Arrangement graphique structuré de caractères, de symboles et d'icônes.
- **Dessin / diagramme (drawing / diagramme)**
*Abstraction spéciale et marquée de signes distinctifs de l'original.
Représentation 2-D, 2 ½ D et 3-D avec une structure topologique similaire à celle de l'original.*
- **Image (picture)**
Représentation qui ressemble fortement à l'original.
- **Monde virtuel (virtual world)**
Environnement 3-D dont l'utilisateur peut faire l'expérience.
- **Réalité augmentée (augmented reality)**
Réalité améliorée par des objets ou des scènes virtuels.

Le quatrième critère est l'**échelle**, qui fait référence à la mesure qui a été choisie comme système de référence géométrique.

- **Aucune**
Il n'y a pas de système géométrique de référence disponible.
- **Linéaire**
Partition équidistante de l'axe.
- **Logarithmique**
Partition logarithmique de l'axe.
- **Exponentiel**
Partition exponentielle de l'axe.
- **Catégorisé**
Partition de l'axe en catégories plutôt que selon une échelle.

Le cinquième critère est le type de **projection géométrique planaire**. Dans ce cadre, une projection est définie comme une transformation d'objets à n dimensions dans un espace avec moins de n dimensions.

- **Aucune**
Pas de besoin pour une projection géométrique planaire.
- **Orthogonale**
Projection avec un centre de projection à distance infinie.
- **Oblique**
Similaire à la projection orthogonale mais avec les rayons de projection qui rencontrent le plan de projection transversalement.
- **Perspective**
Centre de projection à distance finie du plan de projection.

Le sixième critère fait référence à la **dimension temporelle du modèle graphique**. Celle-ci détermine si et comment la visualisation est définie en faisant référence à un axe du temps.

- **Aucun**
Modèle statique sans changement.
- **Discret**
Modèle dynamique qui change à des points discrets du temps.
- **Continu**
Modèle dynamique qui change de manière continue.

Le septième critère est associé au **mode temporel de la présentation**. Celui-ci détermine comment une image unique ou une séquence d'images du modèle graphique est présentée à l'utilisateur, en référence à un axe du temps.

- **Image fixe**
Présentation du modèle (statique ou dynamique) comme une unique image.
- **Vidéo non-proportionnelle**
Présentation de vues discrètes du modèle (statique ou dynamique) comme un flux d'images chronologiquement arrangées.
- **Vidéo proportionnelle**
Présentation du modèle (statique ou dynamique) comme un flux d'images qui sont arrangées chronologiquement et proportionnellement à un axe du temps (vitesse lente, temps réel, vitesse rapide).

Le huitième critère concerne l'**interaction**. Un système est dit interactif quand le temps de réponse du système n'influence pas le processus de travail des utilisateurs.

- **Aucune interaction**
Uniquement regarder la présentation.
- **Navigation dans la présentation**
Contrôler la présentation (ex. Start-Stop).
- **Interaction avec le modèle**
Selon le niveau du modèle, soit changer de vue ou de comportement du modèle.
- **Immersion**
Faire partie du modèle et interagir au sein d'un monde virtuel.

Après avoir proposé ces critères de classification des visualisations d'un point de vue graphique, ils en ont proposé pour distinguer les techniques de visualisation d'un point de vue informationnel.

A ce niveau, le premier critère est relatif à la **fonction primaire** (qu'il appellent aussi l'**intention**) de la technique de visualisation.

- **Identification** (identification)
Détermination de valeurs caractéristiques.
- **Localisation** (localization)
Détermination de positions et de places (relatives ou absolues).
- **Corrélation** (correlation)
Mise en évidence de relations directes.
- **Association** (association)
Mise en évidence de relations indirectes.
- **Comparaison** (comparison)
Mise en évidence de similarités ou de différences.
- **Structures et motifs** (structures and patterns)
Mise en évidence de caractéristiques et de motifs typiques (ex. tendances, hiérarchies).
- **Groupement** (grouping)
Mise en évidence de différentes caractéristiques et des relations correspondantes.
- **Classification** (classification)
Division selon des caractéristiques connues.

Le second critère est le type d'information à représenter.

- **Information quantitative**
Reflète des idées et des corrélations qui ne peuvent pas être décrites et illustrées mathématiquement et sous forme de chiffres.
- **Information qualitative**
Peut être décrite par des quantités.

Le troisième critère fait référence à la **structure relationnelle de l'information** (*information relationship*). Il s'agit du motif formel reconnaissable des corrélations entre des informations individuelles.

- **Indépendante**
- **Relationnelle**
- **Circulaire**

- **Hiérarchique**
- **Réseau**

Le quatrième critère concerne l'**encodage de l'information**.

- **Niveau de mesure** (*level of measurement*)
Indique l'exactitude de l'illustration et en particulier les conclusions déductibles.
- **Dimension de la variable dépendante** (*dimension*)
Nombre de valeurs qui peuvent être codées visuellement pour chaque point d'observation.

Il faut noter que Wenzel et al. précisent que les techniques de visualisation doivent être étudiées pour ce qui est de leur expressivité (*expressiveness*) et de leur efficacité (*efficiency*).

Ils précisent deux buts essentiels d'une visualisation :

- amélioration de la cognition (*cognition enhancement*) ;
- médiation de connaissance (*knowledge mediation*).

La distinction entre 'dimension temporelle du modèle graphique' et 'modèle temporel de la présentation' est importante. De même, il faut distinguer entre la réalité (statique ou dynamique) que l'on veut visualiser, le modèle graphique (statique ou dynamique) utilisé et le mode temporel de la représentation graphique.

Wenzel et al. proposent d'utiliser leur taxonomie pour permettre à un utilisateur de spécifier le but et les exigences de la technique de visualisation dont il a besoin afin d'en déduire et de lui présenter celles qui sont les plus appropriées à son cas. Ce mécanisme peut être implémenté par un programme informatique.

2.2.17. Grinstein, Keim & Ward (2001, 2002, 2004)

Cette taxonomie fut proposée par Keim (2001 ; 2002) pour classifier les techniques d'exploration visuelle de données (*Visual Data Mining*). Elle fut également reprise dans un tutorial relatif à la visualisation d'information et à la recherche visuelle de données dispensé par Grinstein, Keim et Ward au cours de la conférence IEEE Vis'2004.

Grinstein, Keim et Ward proposent une taxonomie selon 3 axes : le **type de données** à visualiser, la **technique de visualisation** et la **technique d'interaction et de distorsion**.

Keim (2002) précise que les trois axes de différenciation sont orthogonaux. De plus il indique qu'un logiciel peut être conçu pour supporter plusieurs types de données différents et combiner différentes techniques de visualisation et d'interaction.

Au niveau des **types de données**, ces auteurs distinguent :

- **Données à une dimension**
- **Données à deux dimensions**
- **Données à plusieurs dimensions**
- **Texte, web**
- **Hiérarchies, graphes**
- **Algorithmes, logiciels**

En ce qui concerne les **techniques de visualisation**, ils proposent les catégories suivantes :

- **Affichage standard 2D / 3D**
- **Affichage géométriquement transformé**
Visualisation de transformations géométriques et de projections des données.
Exemples : matrices de scatterplots, landscapes, coordonnées parallèles, RadViz...
- **Affichage iconique**
Visualisation des valeurs des données sous la forme de caractéristiques d'icônes graphiques (glyphes)
Exemples : faces de Chernoff, stick figures, shape coding, color icons...
- **Affichage dense de pixels (dense pixel display)**
Utilisation de chaque pixel pour visualiser une valeur des données. Les gammes de valeurs des attributs sont mis en correspondance avec une carte de couleurs fixe. Les pixels associés à une dimension sont regroupés dans une même région.
- **Affichage imbriqué (stacked display)**
Visualisation des données en utilisant un partitionnement hiérarchique de l'espace en sous-espaces.
Exemples : Dimensional stacking, treemaps, cone tree, infocube, ...

Enfin, ils distinguent plusieurs **techniques d'interaction et de distorsion** :

- **Standard**
- **Projection**
Variation dynamique ou interactive des projections. Visualisation des paramètres restants en 2 D ou 3 D. Variation automatique des résultats dans une animation.
Exemples : GrandTour, S Plus, Attribute Explorer...
- **Filtrage**
Détermination dynamique ou interactive de sous-ensembles de la base de données.
Exemples : Magic Lenses, Moveable Filter, DeVise, Dynamic Queries...
- **Zoom**
Visualisation de grandes quantités de données sous une forme réduite pour fournir une vue d'ensemble des données. Zoom variable sur les données avec des changements automatiques des modes de visualisation pour présenter plus de détails.
Exemples : Pad++, IVEE, DataSapce...
- **Distorsion**
Distorsion de l'image pour permettre une visualisation de plus grandes quantités de données.
Exemples : Perspective wall, Bifocal displays, TableLens, Graphical fisheye views, Hyperbolic representations, Hyperbox...
- **Link & Brush**
Prérequis : disposer de plusieurs visualisations des mêmes données. Les changements interactifs effectués sur une visualisation sont automatiquement reflétés dans les autres visualisations.
Exemples : Xmdv-tool, S Plus, XGobi, DataDesk...

A titre de commentaire, nous mentionnerons que Grinstein, Keim et Ward reprennent la taxonomie de Grinstein et al. (2001) en précisant que certaines techniques sont des techniques de visualisation pures, d'autres sont intégrées avec de l'analyse des données et enfin les dernières sont des techniques d'interaction.

2.2.18. Rodrigues, Traina, de Oliveira et Traina (2006)

La taxonomie proposée par Rodrigues et al. (2006) se différencie assez fondamentalement de la plupart des autres en plaçant la psychologie cognitive au centre des réflexions. Elle se base sur deux facteurs : le processus de spatialisation et les stimuli pré-attentifs utilisés par les techniques de visualisation. Le processus de spatialisation est défini comme '*la transformation des données d'un format brut difficile à interpréter en un format spatial visible*'. Les stimuli pré-attentifs sont ceux qui correspondent à '*ce qui peut être visuellement identifié avant la phase d'attention consciente*'.

Ces auteurs identifient tout d'abord différentes **méthodes de spatialisation** :

- **Mise en évidence de la structure (structure exposition)**
Techniques de visualisation qui reposent sur des méthodes pour ajuster la présentation des données de manière à ce que leur structure sous-jacente puisse être perçue visuellement.
Exemples : treemaps, graphes basés sur des algorithmes d'équilibre de forces¹⁴
- **Arrangement en motifs (patterned)**
Positionnement le plus simple dans lequel un ensemble de d'éléments individuels sont arrangés séquentiellement (de manière ordonnée ou non) dans une ou plusieurs directions, de manière linéaire, circulaire ou selon des motifs spécifiques.
Exemples : pixel bar charts, camemberts (pie charts), techniques orientées pixels.
- **Projection**
Affichage de données modélisé par la représentation de fonctions. La position d'un élément est définie par une fonction mathématique implicite ou connue.
Exemples : coordonnées parallèles, coordonnées en étoile (star coordinates), graphes 2D ou 3D.
- **Reproduction**
Le positionnement des données est connu d'avance et est déterminé par la disposition spatiale du système dont elles sont issues.
Exemples : carte géographique, représentation mécanique par CAO.

Le second axe de différenciation est relatif aux **stimuli pré-attentifs** :

- **Position**
La position est le premier composant pour la perception pré-attentive et est strictement associée au processus de spatialisation. La pré-attention de la position peut prendre quatre formes :
 - Mise en évidence de la structure → Arrangement
Des arrangements spécifiques décrivent la structure, la hiérarchie ou une autre propriété globale.
 - Arrangement en motifs → Correspondance
La position d'un élément détermine la caractéristique correspondante sans exiger de référence.
 - Projection → Référentiel
Les intervalles de valeurs de la fonction de projection définissent une échelle dans un référentiel permettant une compréhension analogique.
 - Reproduction → Référentiel
La position d'un élément est donnée par rapport à une référence explicite, comme une carte géographique, une forme porteuse de sens ou un ensemble d'axes.

¹⁴ Force directed graph layouts

- **Forme**

Le stimulus pré-attentif lié à la forme offre un grand nombre de possibilités pour exprimer de l'information :

- Différenciation
La forme affichée établit une distinction entre les éléments en vue d'une interprétation postérieure.
Exemples : rectangle de sélection dans un ensemble de points
- Correspondance
Chaque forme perceptible correspond à une caractéristique informationnelle. Elle est discrète si des formes différentes sont utilisées (ex. carré, cercle, croix...) et continue si une forme générique voit certaines de ses propriétés varier (ex. associer des attributs informationnels aux composantes d'un visage qui varient en continu).
- Signification
La forme affichée transmet une sémantique dont la compréhension peut dépendre de la connaissance ou de l'expérience préalable de l'utilisateur.
Exemples : flèche, visage, texte
- Relation
Des formes, telles que des lignes, des contours ou des surfaces, dénotent la relation entre un ensemble d'éléments.
Exemples : coordonnées parallèles, graphes 3D ou chemins en général.

- **Couleur**

La couleur est le troisième stimulus pré-attentif. Elle peut traduire de l'information par deux mécanismes :

- Différenciation
Les couleurs n'ont pas de correspondance spécifique avec les données. Elles renseignent simplement l'égalité ou l'inégalité des certaines caractéristiques.
Exemples : associer une couleur différente à chaque forme en vue d'un double codage (ex. carré rouge, cercle vert, croix bleue...).
- Correspondance
Elle est discrète quand chaque couleur perceptible est mise en correspondance avec une caractéristique informationnelle, comme une classe ou un niveau. Elle est continue quand une échelle de variation de teintes est mise en correspondance avec des données à valeurs continues.

Rodrigues et al. précisent que certaines approches divisent l'espace d'affichage en zones dans lesquelles des processus de spatialisation différents sont appliquées. Ils reconnaissent également que les modalités d'interaction constituent un élément important des techniques de visualisation mais ils ont choisi de ne pas les intégrer dans leur taxonomie. Enfin, ils mettent en évidence le fait que la plupart des taxonomies antérieures manquent de puissance analytique car les constituants de base des techniques ont diffusé dans la structure taxonomique¹⁵. Ils prétendent que leur taxonomie peut s'accomoder de l'émergence de nouvelles techniques car elle repose sur des bases fondamentales communes à toutes les approches.

¹⁵ *They lack analytical power because the core constituents of the techniques are diffused within the taxonomical structure.*

3. Méta-taxonomie des techniques de visualisation

3.1 Introduction

Nous avons montré dans la section précédente qu'un nombre impressionnant de taxonomies des techniques de visualisation ont été proposées dans le domaine de la visualisation d'information. On peut dès lors légitimement s'interroger sur ce qui les distingue. Cette question nécessite d'identifier les propriétés principales permettant de caractériser ces différentes taxonomies. Sur cette base, il sera ensuite possible de positionner les diverses propositions les unes par rapport aux autres, en concevant une méta-taxonomie qui prend en compte les critères de différenciation identifiés.

Avant tout, il faut noter que, dans leurs travaux d'élaboration de taxonomies, certains auteurs ont évoqué la problématique de la multiplicité des classifications mais sans toutefois l'approfondir. Nous commencerons donc notre discussion en rappelant ces éléments de réflexion.

Dans leur article de 1994, Lohse et ses collègues proposent une classification dichotomique en mettant en évidence le fait qu'il existe des taxonomies fonctionnelles et des taxonomies structurales. Les taxonomies fonctionnelles se focalisent sur l'utilisation prévue et sur le but de l'objet graphique. Elles ne reflètent pas la structure physique des images et ne visent pas à correspondre à une représentation sous-jacente dans la mémoire humaine. Les taxonomies structurales se focalisent sur la forme de l'image plutôt que sur son contenu.

Wehrend et Lewis (1990) abordent aussi cette problématique de manière indirecte en précisant qu'il existe des taxonomies destinées à être utilisées par un être humain et d'autres qui peuvent servir de structure à un programme informatique.

Rodrigues et al. (2006) recensent également plusieurs taxonomies proposées dans la littérature¹⁶ et expliquent que la leur s'en différencie en se focalisant sur l'aspect de perception visuelle.

Ces exemples témoignent de la prise de conscience par certains chercheurs du besoin de positionner leur taxonomie par rapport à celles déjà proposées par d'autres. La généralisation de cette approche conduit de manière très naturelle au besoin de définir une méta-taxonomie qui prendrait en compte des axes de différenciation objectifs entre les taxonomies proposées dans la littérature.

3.2 Principes constructifs d'une taxonomie

La construction d'une taxonomie nécessite de préciser deux éléments essentiels : un ensemble de définition et une liste de critères permettant de différencier les éléments de l'ensemble à classer.

Dans le cas qui nous concerne, l'ensemble de définition de la taxonomie semble, de prime abord, facile à identifier. Il s'agit des techniques de visualisation d'information. L'étude des différentes propositions rencontrées dans la littérature nous conduira cependant à nuancer ce propos.

L'étude de l'état de l'art nous a également amenés à observer une grande diversité dans les critères retenus par les différents auteurs pour fonder leur taxonomie. On peut citer, par exemple, le type de données visualisées, les propriétés des éléments graphiques choisis, les tâches supportées par la visualisation... Une première étape dans notre entreprise d'élaboration d'une méta-taxonomie consistera donc à recenser les différents critères de classification rencontrés dans la littérature.

Cette approche met en évidence le fait que les critères de classification peuvent eux aussi faire l'objet d'une taxonomie. En effet, on remarque aisément que tous les critères ne possèdent pas la même capacité à produire des taxonomies précises, complètes et utiles. Il est assez évident

¹⁶ La plupart d'entre elles sont d'ailleurs citées dans ce chapitre.

que l'on peut rencontrer, par exemple, des critères quantitatifs ou qualitatifs, objectifs ou subjectifs, orthogonaux en eux ou dépendants les uns des autres.

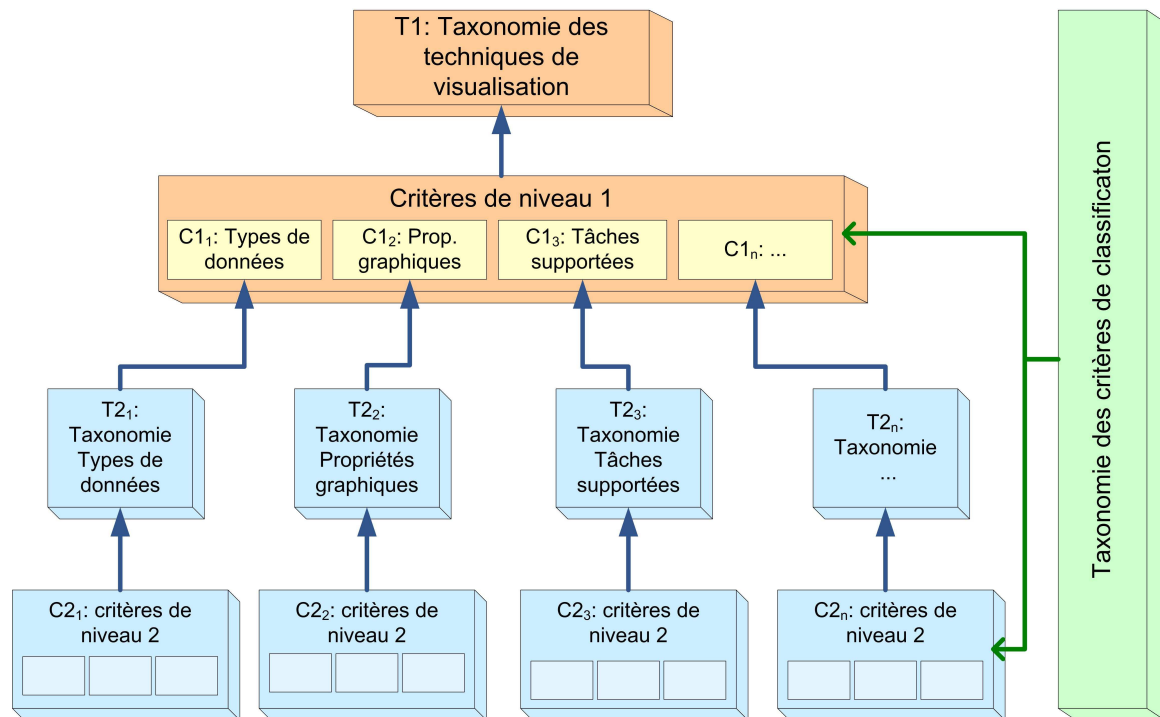


Fig. 13: Vue d'ensemble de la problématique de classification des techniques de visualisation

Si on poursuit la réflexion, on réalise également qu'au niveau de chaque critère, une nouvelle taxonomie doit en principe être définie. En effet, si on souhaite, par exemple, utiliser les types de données comme critère de différenciation des techniques de visualisation, il est nécessaire de disposer d'une taxonomie des types de données et pour construire cette taxonomie de second niveau, il faut aussi disposer de critères de différenciation adéquats. Par exemple, on peut différencier les types de données selon leur dimensionnalité (1-D, 2-D, 3-D) ou leur structure (arbre, graphe, ...). Nous ne considérerons pas les taxonomies de second niveau dans notre travail mais il nous semblait néanmoins utile de faire une remarque à leur propos.

Notre cadre de réflexion général ayant été décrit, nous pouvons à présent aborder l'étude des propriétés des taxonomies de visualisations d'information rencontrées dans la littérature

3.3 Propriétés des taxonomies proposées dans la littérature

Nous commencerons par discuter la problématique du domaine de définition des taxonomies. Ensuite, nous identifierons les critères de différenciation qui ont été utilisés pour distinguer les techniques de visualisation. Nous serons alors en mesure de pouvoir étudier les propriétés de ces critères. Enfin, nous terminerons notre étude par une réflexion sur les méthodologies utilisées par les différents auteurs pour construire leurs taxonomies.

3.3.1. Domaine de définition

a. Domaines de définition des taxonomies proposées dans la littérature

Dans notre démarche de comparaison des taxonomies, la première étape consiste à en préciser les domaines de définition respectifs, c.-à-d. à identifier les objets sur lesquels elles portent.

- Bertin (1998) étudie la construction de représentations graphiques statiques à deux dimensions. Il ne s'intéresse qu'à ce qui est représentable '*sur une feuille plane de papier blanc d'un format moyen, sous un éclairage normal*'. Il exclut explicitement les représentations graphiques incluant du mouvement, de même que les représentations tridimensionnelles.
- Tufte (2001) propose des méthodes de conception des graphiques statistiques.
- La taxonomie de Roth et Mattis (1990) concerne les caractéristiques qui doivent être communiquées à un concepteur humain ou à un ordinateur pour leur permettre de créer des présentations de manière indépendante d'une application informatique. Ils se sont concentrés sur la présentation de données relationnelles en utilisant des variations et synthèses d'affichages 2-D statiques.
- Robertson (1990) limite son étude à la représentation de variables scalaires définies sur un espace à 2 dimensions. De plus, il se focalise sur la technique de représentation par scènes naturelles.
- Werhend et Lewis (1990) ont pour but de proposer une classification des techniques de visualisation qui soit indépendante de domaines d'application particuliers. Ils précisent néanmoins que leur taxonomie est destinée à être utilisée par un être humain et pas par un programme informatique.
- Gruia – Catalin & Cox (1992) se focalisent sur les visualisations de programmes informatiques.
- Lohse et ses collègues (1990, 1994) ont pour but de proposer une taxonomie structurale des représentations visuelles. La gamme des représentations concernées est très large puisqu'elle inclut non seulement les techniques de visualisation de données abstraites, telles que les graphes bidimensionnels, mais aussi les schémas techniques, les logos ou les images photo-réalistes.
- Shneiderman (1996) propose une taxonomie des techniques de visualisations d'information en partant de l'hypothèse qu'il faut afficher aux utilisateurs des collections d'éléments, dans lesquelles les éléments possèdent de multiples attributs.
- Card & Mackinlay (1997) présentent une structuration de l'espace de conception des techniques de visualisation sans préciser de limites explicites aux types de techniques concernées.
- Tweedie (1997) a pour but de caractériser les visualisations interactives.
- Chi (2000) présente une taxonomie des techniques de visualisation sans préciser de limites explicites aux types de techniques concernées.
- Grinstein et ses collègues (2001, 2002) s'intéressent aux visualisations à haute dimension (*high dimensional visualisations*) et plus particulièrement aux '*visualisations qui peuvent présenter un grand nombre de dimensions ou de paramètres relatifs aux données*'. Ils précisent également qu'ils se focalisent sur les

tables de données numériques bien que beaucoup de techniques parmi celles qu'ils citent s'appliquent aussi aux données catégorielles (*categorical data*).

- Keim, Grinstein et Ward (2001, 2002, 2004) proposent une taxonomie des techniques utilisées en fouille visuelle de données (*Visual Data Mining*).
- Tory & Möller (2002) présentent une taxonomie des techniques de visualisation (qu'ils nomment aussi algorithmes de visualisation) sans se limiter explicitement à un domaine de définition.
- Tilley & Huang (2002) se focalisent sur les techniques de visualisation de programmes informatiques.
- Pfitzner et ses collègues (2003) ont pour but de développer une taxonomie complète applicable au paradigme de la visualisation d'information en général.
- Wenzel et ses collègues (2003) proposent une taxonomie des techniques de visualisation utilisées dans la simulation en production et logistique.
- Rodrigues et ses collègues (2006) ne précisent pas a priori de limites aux techniques de visualisation considérées dans leur taxonomie.

b. Catégorisation des domaines de définition

Les taxonomies proposées se différencient donc assez clairement au niveau de leur domaine de définition.

Dans une première catégorie, on peut distinguer celles qui visent, plus ou moins explicitement, à structurer tout l'espace des techniques de visualisation d'information. Cet espace fut défini de manière très générale par Card, Mackinlay et Shneiderman (1999, pp. 6-7) comme '*l'utilisation de représentations visuelles interactives de données abstraites non basées sur des objets physiques dans le but d'amplifier la cognition*¹⁷'.

Différentes taxonomies peuvent être classées dans cette première catégorie, comme celles proposées par Card & Mackinlay (1997), Shneiderman (1996), Pfitzner et al. (2003), Tory & Möller (2002), Chi (2000) ou encore Rodrigues et al (2006). Ces auteurs ne précisent cependant pas tous s'ils adhèrent à la définition précitée pour le domaine tout entier de la visualisation d'information.

Dans une seconde catégorie, on placera les taxonomies qui s'auto-limitent à un sous-ensemble plus ou moins bien délimité des techniques de visualisation d'information. Dans ce cadre, il est intéressant de remarquer que les critères utilisés pour délimiter ces sous-espaces appartiennent à des niveaux conceptuels différents.

- Certains domaines de définition sont définis par des **caractéristiques des données**. Par exemple, Robertson (1990) précise que sa proposition se limite aux représentations de variables scalaires définies sur un espace à deux dimensions.
- Certaines taxonomies délimitent leur portée par rapport à des **propriétés des techniques de visualisation** elles-mêmes. On peut citer par exemple les dimensions disponibles pour la représentation graphique (cf. Bertin, 1998) ou la présence d'interactivité (cf. Tweedie, 1997).
- Il existe également des taxonomies qui se focalisent sur un **type d'application** particulier, tel que l'exploration visuelle de données (cf. Grinstein et al., 2004 ; Keim, 2002).
- Enfin d'autres se limitent à un **secteur d'activité** spécifique tel que la visualisation de programmes informatiques (cf. Gruia-Catalin et Cox, 1992 ; Tilley & Huang, 2002) ou la simulation en production et logistique (cf. Wenzel et al., 2003).

¹⁷ 'Use of interactive visual representations of abstract, non physically based data to amplify cognition.'

En conclusion, nous pouvons donc retenir deux éléments principaux quand on souhaite définir une taxonomie. D'une part, il est indispensable d'identifier clairement son domaine de définition. D'autre part, ce domaine de définition peut être délimité de différentes manières : caractéristiques de données à visualiser, propriétés des techniques de visualisation, utilisation prévue pour la taxonomie, domaine applicatif...

Nous retiendrons donc un premier principe général : « **Une taxonomie doit préciser son domaine de définition** ».

Si la taxonomie se veut spécifique à un sous-ensemble des techniques de visualisation, il est également indispensable de mentionner les critères qui délimitent ce sous-ensemble. Cette exigence constitue en fait une variante du premier principe : « **Si le domaine de définition se réduit à un sous-espace de l'espace des techniques de visualisation, la taxonomie doit mentionner les critères utilisés pour délimiter ce sous-ensemble** ».

3.3.2. Critères de différenciation des visualisations

L'étude des taxonomies proposées dans la littérature nous indique qu'une large gamme de critères est utilisée pour classer les techniques de visualisation d'information.

En guise d'introduction à cette problématique, il nous semble utile de mentionner le fait que, parmi les concepteurs des taxonomies étudiées, très peu estiment nécessaire de s'attarder sur les critères permettant de juger de la pertinence des critères de classification sous-jacents à leur taxonomie (Roth & Mattis, 1990 ; Tilley & Huang, 2002 adoptent cette approche).

Certaines taxonomies se fondent sur une **classification des données** à visualiser. Plusieurs approches ont été proposées, ainsi que l'illustrent les exemples suivants.

- Card et Mackinklay (1997) distinguent des ensembles de données quantitatifs, ordinaux et nominaux.
- Shneiderman (1996) identifie sept types de données : données 1-D, données 2-D, données 3-D, données temporelles, données multidimensionnelles, arbres et réseau.
- Pfizner et al. (2003) distinguent des données de bas niveau (objet, attribut) et des données de haut niveau (méta-données). Ils identifient aussi cinq types de relations entre les données : linéaire, circulaire, arbre ordonné, arbre non ordonné et hyper-espace.
- Roth et Mattis (1990) ont identifié différents types de structure des données, selon leurs propriétés de couverture relationnelle, de cardinalité et d'unicité.
- Grinstein et al. (2004) ainsi que Keim (2002) recensent six types de données : données à une dimension, données à deux dimensions, données à plusieurs dimensions, texte et web, hiérarchies et graphes, algorithmes et logiciels.

Certains chercheurs se sont basés sur les **propriétés graphiques des techniques de visualisation**, ainsi que l'illustrent les exemples suivants.

- Bertin (1998) classe les représentations graphiques sur base des correspondances qu'elles établissent dans le plan (cf. *groupes d'imposition*) et selon la disposition de leurs composants (cf. *types d'imposition*).
- Tory & Möller (2002) utilisent le nombre de variables (indépendantes et dépendantes) représentées dans la visualisation pour opérer une classification.
- Bertin (1998) propose également une liste des variables visuelles disponibles dans le système graphique : variables du plan (x,y), taille, valeur, grain, couleur, orientation, forme.
- On peut aussi mentionner la distinction qu'opèrent Grinstein et Ward (2002) entre les visualisations ayant un focus géométrique et celles ayant un focus symbolique.

- Pfitzner et al. (2003) distinguent les visualisations animées et non animées selon qu'il y ait ou non du mouvement ou du changement dans la représentation.
- Wenzel et al. (2003) distinguent les visualisations selon leur dimension, le type de représentation, le format d'affichage, l'échelle, le type de projection géométrique planaire, la dimension temporelle du modèle graphique, le mode temporel de la représentation.
- Grinstein et al. (2004) et Keim (2002) distinguent les affichages standard 2D / 3D, les affichages géométriquement transformés, les affichages iconiques, les affichages denses de pixels, les affichages imbriqués.

Les **propriétés cognitives des techniques de visualisation** ont aussi été employées pour sous-tendre une taxonomie.

- Lohse et ses collègues (1994) distinguent, par exemple, des représentations visuelles faciles ou difficiles à comprendre, attractives ou non attractives, accentuant l'ensemble ou les parties, communiquant beaucoup ou peu d'information.
- Card et Mackinley (1997) regroupent les composants graphiques d'une visualisation selon qu'ils sont traités automatiquement ou de manière contrôlée au niveau cognitif.
- Ware (2004) distingue les symboles et les aspects des techniques de visualisation qui sont sensoriels et arbitraires. Les premiers dérivent leur pouvoir d'expression de leur capacité à utiliser la puissance de traitement perceptuel du cerveau sans apprentissage alors que les seconds doivent être appris car ils en possèdent aucune base perceptuelle.
- Rodrigues et al. (2006) adoptent les processus de perception visuelle (stimuli pré-attentifs) comme un des deux axes de classification de leur taxonomie.

On peut également distinguer différents **niveaux d'interactivité**.

- Tweedie (1997) distingue différentes formes d'interactivité : manuelle, mécanisée, enseignable, guidable, automatique.
- Wenzel (2003) identifie quatre niveaux d'interaction : aucune interaction, navigation, interaction, immersion.
- Gruia-Catalin et Cox (1992) distinguent les systèmes de visualisation selon la manière dont ils permettent la spécification des visualisations : prédéfinition, annotation, association, déclaration et manipulation.
- Tilley & Huang (2002) distinguent trois catégories de visualisations sur base du niveau d'interaction entre l'utilisateur et la représentation graphique : visualisations statiques, interactives et éditables.
- Pfitzner et al. (2003) distinguent les affichages statiques et dynamiques selon qu'ils permettent ou non de l'interaction.

Le **niveau d'abstraction** peut également constituer un critère de différenciation. On fait ici référence à la relation existant entre les données brutes à visualiser et les informations représentées graphiquement.

- Gruia-Catalin et Cox (1992) identifient cinq niveaux d'abstraction : représentation directe, structurale, synthétisée, analytique et explicative.
- Tweedie (1997) fait également référence à la différence entre les données brutes et l'information qui est visualisée en parlant de 'représentations de données construites' et de 'représentations de données converties'.

- Tory & Möller (2002) basent leur taxonomie sur la distinction à opérer entre les données et le modèle de ces données qui sous-tend la visualisation. Au niveau le plus haut, ils distinguent les modèles de données continus et discrets.

Les **tâches que l'utilisateur peut effectuer** sur elles ont également été utilisées pour classer les techniques de visualisation.

- Shneiderman (1996) identifie sept tâches fondamentales : vue d'ensemble, zoom, filtre, détails à la demande, mise en relation, historisation, extraction.
- Grinstein, Keim et Ward (2004) parlent en termes de techniques d'interaction et de distorsion mais il s'agit plutôt d'un mélange entre tâches et réelles techniques d'interaction. Ils en identifient six types : standard, projection, filtrage, zoom, distortion, *link & brush*.

Les techniques de visualisations d'information peuvent aussi être classées selon **leurs buts**, c.-à-d. d'après les questions sur les données auxquelles elles permettent de répondre. Cet axe de classification présente un chevauchement conceptuel assez marqué avec l'axe de classification basé sur les tâches.

- Bertin (1998) propose une approche en deux temps. Tout d'abord, il indique qu'il y a autant de types de questions que de composantes dans une information. Ensuite, il distingue 3 niveaux de lecture : le niveau élémentaire de lecture, des niveaux moyens de lecture et le niveau d'ensemble.
- Roth et Mattis (1990) distinguent des buts associés à la fonction de présentation des données et d'autres associés à la distribution des données dans la présentation.
- Robertson (1990) adopte une position similaire en précisant qu'une représentation graphique peut avoir plusieurs buts aux yeux de l'analyste des données : la valeur d'une variable en un point, la distribution locale des valeurs (gradients, caractéristiques) ou à la distribution globale des valeurs (tendances, structure). De plus, l'analyste peut s'intéresser à une seule variable ou à la comparaison de plusieurs variables, à des niveaux éventuellement distincts.
- Wehrend et Lewis (1990) proposent eux aussi une liste des buts possibles sous-tendant l'utilisation de la visualisation : identifier, localiser, distinguer, catégoriser, regrouper, distribuer, classer, comparer, étudier les relations entre et au sein de, associer et corrélérer.
- Wenzel et al. (2003) mentionnent une liste de buts (*'fonctions primaires'*) possibles pour une visualisation : identification, localisation, corrélation, association, comparaison, structure et motifs, groupement et classification.

Les **fonctions générales de la technique de visualisation** peuvent aussi servir de base à leur catégorisation.

- Bertin (1998) distingue trois fonctions possibles pour une représentation graphique : enregistrer de l'information, communiquer de l'information ou traiter de l'information.
- Wenzel et al. (2003) proposent une liste d'utilisations possibles de visualisations : outil d'acquisition, utilitaire de vérification, utilitaire de validation, outil d'analyse, outil de communication, outil de présentation, outil de formation.
- Grinstein, Keim et Ward (2004) regroupent les visualisations à haute dimension en trois groupes : les techniques de visualisation pures, les techniques de visualisation intégrées avec des techniques d'analyse de données et les techniques de visualisation associées à des interactions.

Une taxonomie peut aussi avoir pour but de classifier les types de **processus de construction d'une visualisation**.

- La taxonomie de Chi (2000) se base sur le modèle d'état des données, qui différencie les techniques de visualisation selon trois types d'éléments constitutifs : les phases des données, les opérateurs de transformation des données et les opérateurs internes aux phases.
- Pour Rodrigues et al. (2006), le processus de spatialisation constitue un axe des deux axes de différenciation des techniques de visualisation. Il distingue : la mise en évidence de la structure, l'arrangement en motifs, la projection et la reproduction.

Certaines taxonomies sont parfois aussi délimitées en **précisant l'utilisation** qui peut en être faite.

- Werhend & Lewis (1990) indiquent que leur taxonomie est prévue pour être utilisée par un être humain.
- Roth et Mattis (1990) destinent leur classification indifféremment à un être humain ou à un programme informatique.

3.3.3. Remarque sur la démarche de construction des taxonomies

En guise de remarque, la démarche utilisée par l'auteur de la taxonomie pour la construire mérite également d'être discutée succinctement. On observe deux grands types de processus de construction des taxonomies. Dans la plupart des cas, la taxonomie résulte d'une réflexion de nature plutôt théorique basée sur un état de l'art des techniques proposées dans la littérature. Dans un nombre limité de cas, la taxonomie est le fruit d'études expérimentales spécifiquement conçues à cette fin (cf. p.ex. Lohse, 1990).

a. Taxonomies issues de réflexions théoriques sur un état de l'art

La plupart des taxonomies résultent de réflexions sur les propriétés que possèdent les différentes techniques de visualisation proposées dans la littérature. Selon les cas, un état de l'art est utilisé comme matière première à la réflexion ou comme moyen de validation de la taxonomie proposée.

b. Taxonomies issues d'expériences

Les classifications de représentations visuelles de Lohse et al. (1990, 1994) constituent un des rares exemples de taxonomies résultant d'études expérimentales.

La méthodologie consiste essentiellement à demander à des sujets de caractériser des exemples de visualisations d'information en leur attribuant des valeurs par rapport à des critères pré-établis (p.ex. 'facile à comprendre - difficile à comprendre'). Puis, par une opération statistique sur les résultats, des groupes de visualisations ('clusters') sont construits. Ils sont supposés refléter la classification que les utilisateurs opèrent intuitivement.

3.3.4. Propriétés des critères de différenciation

Nous avons montré précédemment à quel point peuvent varier les critères utilisés pour classer les techniques de visualisation d'information. Il nous semble important d'étudier plus en détail les propriétés de ces critères en tant qu'outils de classification. En effet, l'utilisation de critères plus ou moins adaptés ou efficaces a un effet direct sur la qualité de la taxonomie qui en découle.

Les deux premières propriétés (univocité et valeurs disjointes) font référence à la qualité du critère au niveau sémantique. La troisième (complétude) établit un rapport entre le critère et l'espace de définition de la taxonomie. Les quatrième et cinquième propriétés (pouvoir discriminant, valeur unique) précisent les cardinalités des relations entre les techniques de visualisation et les valeurs du critère. Enfin, la dernière propriété (orthogonalité) porte sur la relation entre les critères.

a. *Univocité*

Le critère de différenciation sera qualifié d'*univoque* s'il est défini de manière à éviter les interprétations ambiguës. Par exemple, le critère '*facile ou difficile à comprendre*' est clairement non univoque alors que le critère de classification portant sur l'utilisation de l'une ou l'autre variable du système graphique de Bertin est non ambigu (p.ex. classer les techniques de visualisation selon qu'elles utilisent ou non la variable graphique 'couleur').

b. *Valeurs disjointes*

Un critère de différenciation est '*à valeurs disjointes*' si les définitions des valeurs qu'il peut prendre ne présentent pas de chevauchement sémantique. Cette propriété ne concerne pas le nombre de valeurs du critère qu'une technique de visualisation peut se voir attribuer mais le fait que ces valeurs soient clairement différentes.

Par exemple, un critère qui permettrait les valeurs 'données hiérarchiques' et 'données hiérarchiques temporelles' ne posséderait pas cette propriété car une technique de visualisation de données hiérarchiques temporelles est aussi une technique de visualisation de données hiérarchiques.

c. *Complétude*

Nous définirons la complétude d'un critère de classification comme suit. Elle consiste à évaluer si les valeurs proposées dans la taxonomie pour le critère de différenciation concerné couvrent toute la gamme des valeurs connues ou possibles pour ce critère dans le domaine de définition de la taxonomie.

Par exemple, si on utilise le critère de différenciation '*types de données*'. On le qualifiera de '*complet*' si tous les types de données connus ou possibles dans le domaine de définition de la taxonomie sont des valeurs acceptées.

Par contre, une énumération de techniques de visualisation ne peut, par définition, pas être un critère complet car il suffit qu'une nouvelle technique soit proposée pour devoir l'ajouter aux valeurs possibles.

Il est intéressant de noter que dans certains cas, il est impossible de déterminer si le critère est complet ou non. En effet, on ne peut affirmer qu'un critère de différenciation est complet que si on dispose de la liste de toutes les valeurs possibles ou d'un moyen pour la construire. Or, il arrive que cette liste soit impossible à produire. Par exemple, la liste des techniques d'interaction peut à tout moment être élargie par un chercheur proposant une nouvelle approche révolutionnaire.

d. Pouvoir discriminant

Nous appellerons '*pouvoir discriminant*' le fait que chaque élément à classer possède (au moins) une valeur non nulle pour le critère de différenciation concerné. Cette propriété définit la cardinalité minimale de la relation entre une technique de visualisation et les valeurs du critère de différenciation.

Un critère '*discriminant*' permet de distinguer tous les éléments de l'ensemble en évitant les problèmes de valeurs nulles ou indéterminées.

Par exemple, le critère '*présence d'animation dans la visualisation*' est discriminant si on peut évaluer, pour toute technique de visualisation incluse dans le domaine de définition, si elle inclut ou non du mouvement (c.-à-d. de l'animation).

e. Valeur unique

On dira qu'un critère de définition est '*à valeur unique*' si toute technique de visualisation ne peut prendre au plus qu'une seule valeur pour le critère donné. Cette propriété définit la cardinalité maximale de la relation entre une technique de visualisation et les valeurs du critère de différenciation.

Par exemple, le critère '*type de données*' de Keim (2002) est à valeur unique car une donnée ne peut simultanément prendre plusieurs valeurs dans la liste : 'données 1-D', 'données 2-D', 'données à plusieurs dimensions', 'texte et web', 'hiérarchies et graphes', 'algorithmes et logiciels'.

Comme contre-exemple, on peut citer le cas du critère relatif aux stimuli pré-attentifs dans la taxonomie de Rodrigues et al. (2006). Une même technique de visualisation peut utiliser simultanément les stimuli basés sur la 'forme' et sur la 'couleur'.

f. Orthogonalité

Une remarque s'impose d'emblée pour cette propriété. A la différence des propriétés précédentes qui sont définies par rapport à un seul critère (p.ex. complétude), l'orthogonalité concerne la relation entre plusieurs critères de classification.

En effet, nous dirons que n critères de classification sont *orthogonaux* s'ils sont indépendants entre eux. En d'autres termes, les valeurs disponibles pour chaque critère ne contiennent pas de dépendance par rapport aux valeurs des autres critères. A chaque valeur d'un axe de classification donné, peut être associée toute valeur de tout autre axe. Naturellement, il se peut qu'en pratique aucune technique de visualisation n'ait (encore) exploité cette combinaison de valeurs.

Par exemple, le caractère animé ou inanimé de la visualisation est indépendant du type de données visualisées (p.ex. 1-D, 2-D, arbres, graphes...).

3.3.5. Qualités d'une bonne taxonomie

Nous résumerons notre réflexion sur les taxonomies étudiées en explicitant les qualités qui, selon nous, doivent être présentes dans une bonne taxonomie. Une 'bonne' taxonomie doit offrir au moins les deux propriétés suivantes :

- elle est rigoureusement construite selon un processus explicite ;
- elle est relativement stable, dans le sens où seules des avancées majeures dans son domaine de définition conduisent à en modifier la structure fondamentale.

Pour atteindre ces objectifs, elle doit remplir différentes conditions présentées ci-après.

- Son domaine de définition doit être précisément défini.
- Les critères utilisés pour classifier les éléments de son ensemble de définition doivent être *univoques*, à *valeurs disjointes*, *discriminants* et *orthogonaux entre eux*. Dans la mesure du possible, ils doivent également être *complets*.

Le fait qu'elle repose sur des critères à *valeur unique* n'est pas indispensable pour autant que les autres propriétés soient satisfaites. En effet, dans certains cas, il est naturel que le critère permette l'attribution simultanée de plusieurs valeurs. Par exemple, si les variables rétinienne de Bertin sont utilisées comme critère de classification, il est évident qu'une technique de visualisation peut en utiliser plusieurs (p.ex. couleur et forme).

Il faut noter que le nombre de critères de classification ne constitue pas en soi un gage de qualité. Nous le considérons plutôt comme une dimension de la taxonomie. Par exemple, une taxonomie de dimension 'un' qui précise uniquement les dimensions de l'espace d'affichage peut être de très bonne qualité. Elle aura simplement un pouvoir descriptif moindre qu'une taxonomie de dimension 'deux' qui ajoute à ce critère le type de données représentées. Un des défis principaux à surmonter lors de la conception d'une taxonomie des techniques de visualisation consiste donc dans le choix d'un nombre suffisant de dimensions sans altérer la qualité intrinsèque de la taxonomie.

Sur base de notre étude des différentes propositions de taxonomie rencontrées dans la littérature, nous constatons que construire une taxonomie de haute qualité est un défi difficile à relever. Plusieurs taxonomies combinent en effet des axes de classification rigoureusement construits avec d'autres dont la conception est plus légère. De plus, plusieurs taxonomies ne discutent pas suffisamment leur domaine de définition. Certains auteurs, tels Grinstein et Ward (2002) avouent d'ailleurs explicitement les limites de leur proposition en affirmant que toutes les techniques de visualisation ne se retrouvent pas précisément dans la classification et que les lignes de séparation ne sont pas précises.

En conclusion, compte tenu de l'importance qu'une taxonomie de bonne qualité peut présenter en termes de support à la conception de techniques de visualisation, de comparaison de solutions, d'identification de voies nouvelles de recherche ou encore de formation à ce domaine scientifique, nous pensons qu'un démarche rigoureuse d'unification des taxonomies proposées depuis une vingtaine d'années pourrait s'avérer bénéfique pour la discipline de la visualisation d'information.

Nous ne consacrerons cependant pas notre recherche à cette vaste entreprise. Nous nous limiterons à proposer une taxonomie de portée plus restreinte, focalisée sur les techniques permettant de visualiser la coopération électronique. Cette taxonomie sera décrite dans un chapitre ultérieur (cf. p. 125).

Chapitre 4 : Visualisation et coopération

« Le dessin, que d'un autre nom nous appelons trait, est ce en quoi consiste et ce qui constitue la source et le corps de la peinture, de l'architecture et de tous les autres genres d'art, et la racine de toutes les sciences. »

Michel-Ange

De multiples approches ont été proposées pour visualiser des données associées à un contexte de travail coopératif. Elles peuvent être distinguées selon différents axes d'analyse : le type de données visualisées, le type de technique de visualisation, le nombre de dimensions des données ou des visualisations, le type d'interaction avec la visualisation...

Certains experts en visualisation d'information, comme Bertin, prônent de se concentrer en premier lieu sur les données à représenter pour déceler ensuite les techniques les plus adaptées. Nous adopterons ce point de vue pour notre état de l'art des techniques de visualisation associées au travail coopératif car nous pensons également qu'il est primordial de connaître les propriétés et caractéristiques des données avant de concevoir de nouvelles techniques de visualisation.

L'étude de l'état de l'art nous indique que des données de natures différentes peuvent être visualisées pour supporter un environnement coopératif. Nous avons recensé les catégories de données suivantes. On peut visualiser :

- des communications (le plus souvent électroniques),
- des actions sur des ressources partagées,
- la présence d'acteurs dans un environnement partagé,
- des événements présentant un intérêt collectif.

Chacune de ces catégories est détaillée dans les sections suivantes.

1. Visualisations de communications

Un courant de recherche assez important se focalise sur la représentation graphique de communications. A ce niveau, le média de communication nous semble particulièrement adéquat comme critère de différenciation de ces recherches. Il faut noter que dans la plupart des cas, les conversations visualisées reposent sur l'utilisation de média de communication électroniques. Nous présenterons donc une revue de travaux ayant pour but de visualiser des communications par messagerie électronique (*e-mail*), par messagerie instantanée (*chat*, *instant messaging*) et par groupes de discussions (*newsgroup*, *forums*, *message boards*).

1.1 Communications par messagerie électronique

Le premier groupe de travaux et probablement le plus important en termes de volume concerne la visualisation de conversations par messagerie électronique (*e-mail*). Ce média est particulièrement étudié car il s'agit de l'outil électronique le plus utilisé en pratique pour supporter la coopération. En fait, diverses études ont montré que cet outil initialement destiné à de la simple communication a vu son usage détourné pour devenir un moyen de gérer le temps et les tâches (cf. p.ex. les travaux séminaux de Mackay, 1988 et de Whittaker et Sidner, 1996), à un tel point que Ducheneaut et Bellotti (2001) parlent de la messagerie électronique comme d'un habitat électronique.

Plusieurs approches ont donc été explorées sur le thème de la visualisation de messages électroniques. Viégas, Golder et Donath (2006) précisent que les travaux de recherche sur cette question peuvent être classés en quatre grandes catégories :

- les visualisations basées sur des chaînes de messages (*threads-based representations*),
- les visualisations liées aux réseaux sociaux (*social-network visualizations*),

- les visualisations temporelles (*temporal visualizations*) et
- les visualisations basées sur les personnes de contact (*contact-based visualizations*).

Nous reprendrons cette classification en y ajoutant une catégorie regroupant des travaux récents proposant des visualisations focalisées sur les tâches (*task-based visualizations*).

Un premier groupe de travaux se concentre donc sur la **représentation graphique de la structure de conversations** (c.-à-d. de chaînes de messages reliés par des opérateurs de type ‘Reply’ ou ‘Forward’). On peut mentionner comme exemple bien connu les travaux du projet de recherche d’IBM intitulé ‘ReMail’. Rohall et al. (2001) ont ainsi proposé une représentation sous forme arborescente combinée avec une ligne du temps non linéaire qui comprime les périodes avec peu d’activité. Le prototype ‘Tread Arcs’ a été développée par Kerr (2003) dans le cadre du même projet (cf. Fig. 14). La représentation intègre de l’interactivité qui permet de mettre en valeur des propriétés de la chaîne complète ou de messages individuels. Par exemple, il est possible de montrer l’importance d’un message particulier de la chaîne (cf. Fig. 14 a). On peut également mettre en évidence des propriétés spécifiques, comme la contribution de chaque personne. (cf. Fig. 14 b).

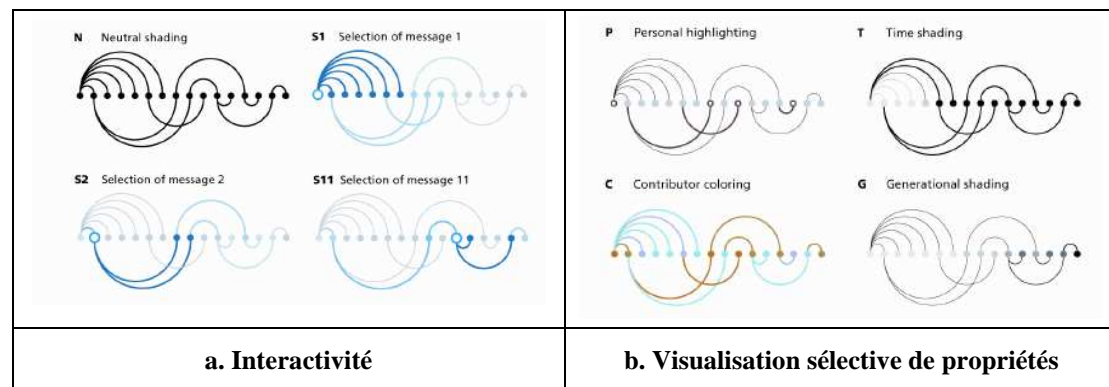


Fig. 14: Représentation ‘Thread Arcs’ (source: [104])

A titre d’autre exemple qui n’est pas associé au projet ‘ReMail’, citons ‘VisuThread’ proposée par Otjacques et Feltz (2004) qui utilise une visualisation semblable aux diagrammes d’interactions UML (cf. Fig. 15).

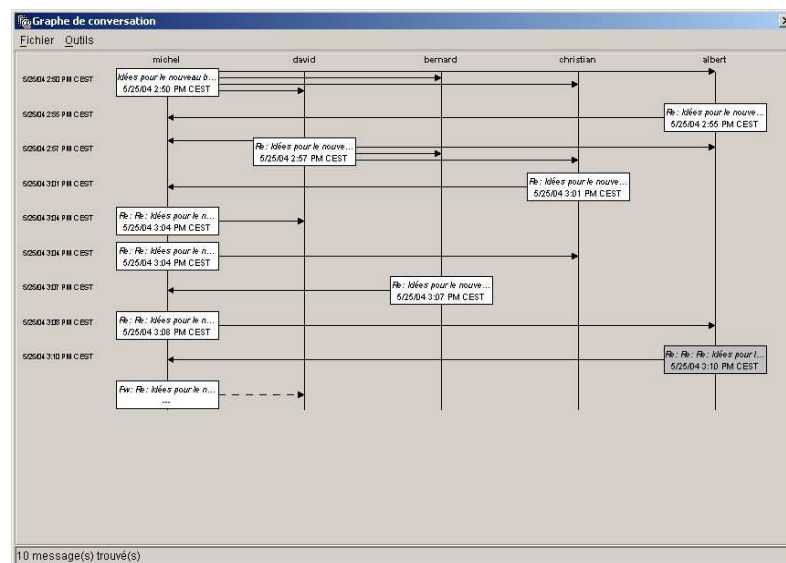


Fig. 15: Représentation ‘VisuThread’ (source : [145])

Dans cet exemple, le temps est associé à l'axe vertical et les intervenants à l'axe horizontal. Une originalité de cette approche réside dans le fait que des messages futurs sont représentés. Cette fonctionnalité est implémentée en visualisant les messages placés dans le répertoire 'Draft' sous la forme d'une flèche en pointillés (cf. Fig. 15, dernier message en bas du graphique). Comme ces messages ne possèdent pas de date d'envoi, ils sont simplement positionnés dans le futur sans référence temporelle absolue (date explicite) mais en utilisant néanmoins une référence relative au temps (*'se situe après l'envoi du dernier message représenté'*).

On peut également mentionner le prototype 'Mat'Graph' (2005) basé sur l'utilisation combinée d'une représentation matricielle et d'arc orientés (flèches) pour représenter la directionnalité des messages. Cette approche mixte vise à trouver un compromis entre les techniques de visualisation par graphe et par matrice.

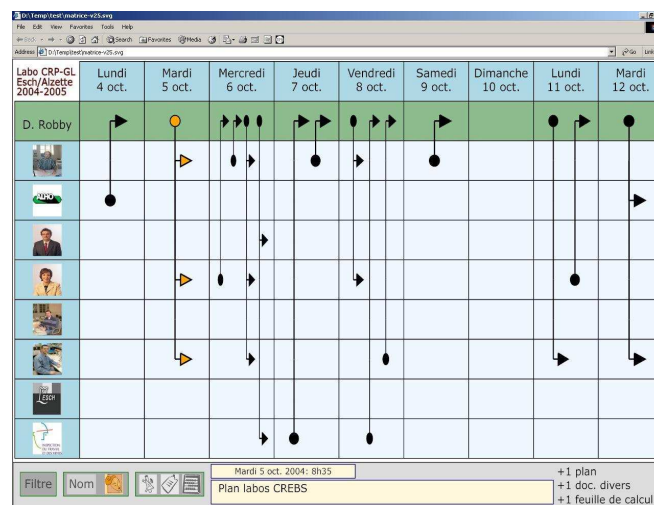


Fig. 16: Visualisation 'Mat'Graph' (source: [146])

Le second groupe de travaux vise à **représenter graphiquement la structure des réseaux sociaux supportés par la messagerie électronique**. Ces visualisations ne sont pas dédiées à la représentation de messages individuels telles que celles présentées précédemment mais proposent des vues plus ou moins agrégées des messages.

Une proportion importante de ces travaux repose sur la théorie de l'analyse des réseaux sociaux (*Social Network Analysis – SNA*, cf. Wasserman et Faust, 1999 pour une introduction à cette théorie) qui étudie la signification sociale de propriétés mathématiques du graphe des relations entre des personnes. La théorie SNA a, par exemple, été utilisé pour étudier la structure des communications au cours de projets de construction (Howard et Petersen, 2001). Les représentations associées se présentent habituellement sous la forme de graphes (cf. Fig. 17).

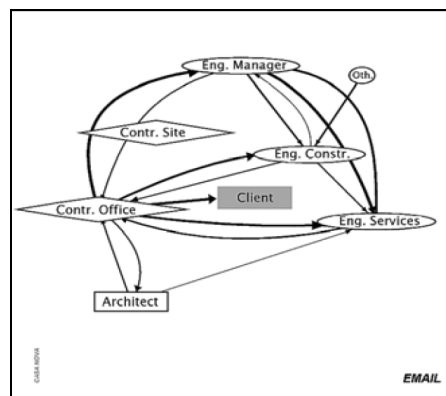


Fig. 17: Représentation classique associée à la théorie SNA (source : [94])

Issus de travaux conduits dans le groupe ‘*Sociable Media Group*’ du MIT (Boyd, 2002 ; Boyd et Potter, 2003), l’approche ‘*Social Network Fragments*’ a pour but de permettre à l’utilisateur de naviguer dans les structures de réseaux sociaux qui émergent au cours de conversation par courriels. Sur base de l’analyse des expéditeurs et des destinataires, l’outil présente le graphe des connections interpersonnelles (cf. Fig. 18, partie gauche de l’interface graphique). Le graphe révèle des groupes de personnes qui sont fortement connectés. La partie droite de l’interface graphique présente l’historique des messages sous une forme matricielle. La composante dynamique est aussi intégrée par le biais d’animations qui montrent comment les relations émergent au cours du temps.

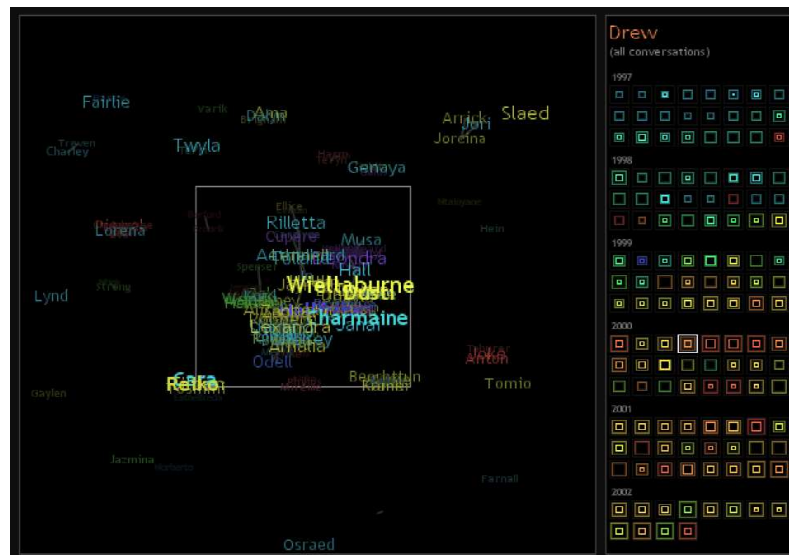


Fig. 18: Représentation ‘*Social Network Fragments*’ (source: [29])

Fisher et Dourish (2004) ont également proposé un ensemble d’outils visuels (appelé ‘*Soylent*’) destinés à analyser les archives de messages électroniques dans une vue égo-centrique. La vue ‘*Network View*’ visualise les réseaux sociaux sous forme de graphes.

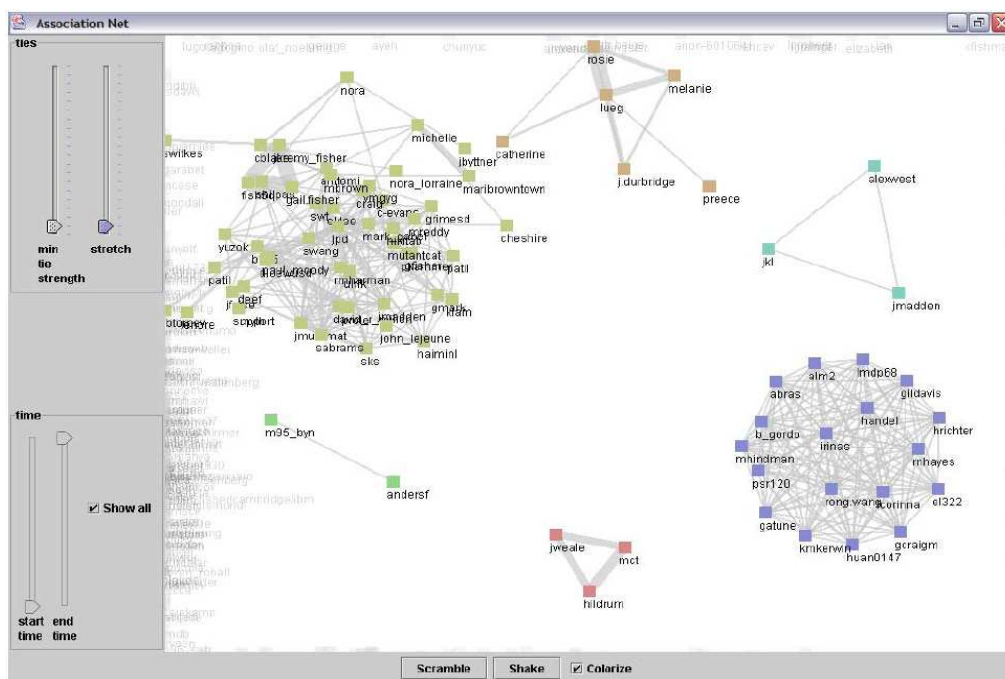


Fig. 19: Vue ‘*Network View*’ de la suite ‘*Soylent*’ (source: [66])

La technique ‘*Clique Panel*’ (Wei-Jen et al., 2004) est un autre exemple, moins habituel, de visualisation de structures sociales basées sur la messagerie électronique. Elle consiste à visualiser des ‘*cliques*’ c.-à-d. des groupes de personnes (identifiées par leur compte de messagerie) qui participent les uns avec les autres dans des communications communes. Ses auteurs identifient deux types de cliques : celles qui sont basées sur l’analyse des archives d’un compte de messagerie d’un seul utilisateur (*user cliques*) et celles qui émergent de l’étude des flux entre un groupe d’utilisateur (*enclave cliques*).

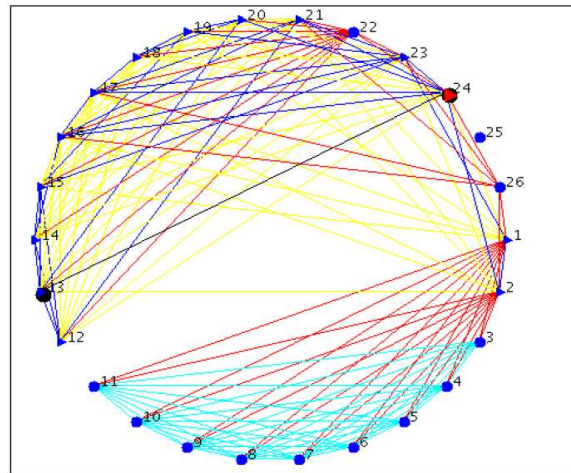


Fig. 20: Visualisation ‘*Clique Panel*’ (source: [117])

Dans la visualisation (cf. Fig. 20), les neuds positionnés sur la périphérie du cercle représentent les cliques. Plus précisément, les nœuds sont des polygones dont le nombre de côtés indique le nombre de membres dans la clique. Si deux cliques ont un membre en commun, elles sont reliées par un lien dans la visualisation. La couleur de cette droite indique le pourcentage de membres identiques dans les deux cliques concernées, ce qui met en évidence de degré de similarité des cliques. Cette approche se différencie fondamentalement des deux précédentes car elle ne visualise pas les utilisateurs ou les messages électroniques mais des informations dérivées de ces données de base.

Un troisième ensemble de travaux se focalise sur la **visualisation de la temporalité des messages** (c.-à-d. la chronémique – *chronemics*¹⁸). La technique classique rencontrée dans ce contexte est la ligne du temps. Elle sert d’ailleurs de référence à la structure de liste utilisée par la plupart des clients de messagerie électronique. Néanmoins, certains chercheurs ont aussi utilisé une visualisation de type matricielle dans laquelle un axe est associé au temps et l’autre aux personnes (p.ex. prototype *TimeStore*, Yiu et al., 1997). À côté de ces méthodes visant à représenter individuellement les messages, d’autres approches se basent sur des données agrégées. Par exemple, Fisher et Dourish (2004) proposent une vue qui affiche, pour chaque mois, les dix contacts auxquels l’utilisateur courant a envoyé le plus de messages électroniques (cf. Fig. 21).

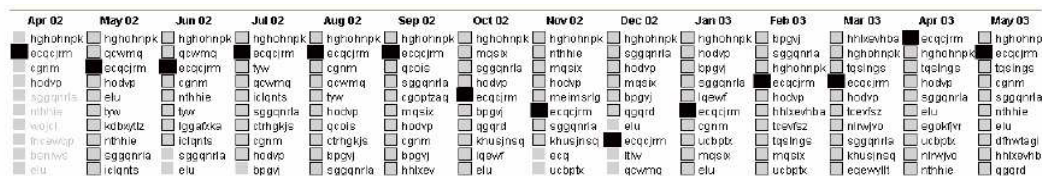


Fig. 21: Vue ‘*Monthly Top Ten*’ de la suite ‘*Soylent*’ (source: [66])

¹⁸ *Chronemics refers to the temporal dimension of communication* (cf. [205], p. 980).

Le prototype ‘*Themail*’ (Viégas et al., 2006) propose également une approche très originale en représentant la distribution au cours du temps des mots les plus utilisés dans le contenu des messages échangés entre l'utilisateur et un de ses contacts (approche dyadique). Le temps est associé à l'axe horizontal et des colonnes de mots sont affichées en utilisant différentes couleurs et tailles selon la fréquence et le caractère distinctif des mots (cf. Fig. 22). Cette représentation vise à mettre en évidence les thèmes les plus discutés avec une personne donnée au cours d'une certaine période.



Fig. 22: Représentation ‘*Themail*’ (source: [205])

Certains chercheurs ont aussi étudié comment combiner les deux approches focalisées sur la structure des messages et la temporalité des échanges. Par exemple, Venolia et Neustaedter (2003) ont combiné une liste séquentielle des messages avec une représentation arborescente des conversations.

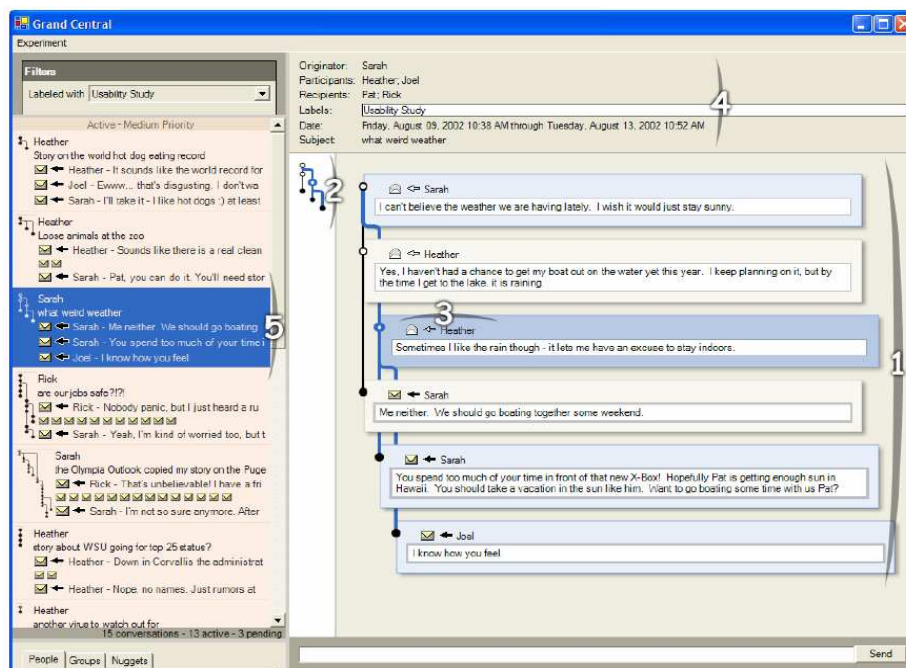


Fig. 23: Visualisation mixte de Venolia et Neustaedter (source: [204])

Le quatrième groupe de **visualisations** associées à des messages électroniques rassemble celles **qui placent les personnes de contact au centre des préoccupations**. L'exemple le plus connu de cette approche est sans doute le prototype '*Contact Map*' (Nardi et al., 2002) (cf. Fig. 24).

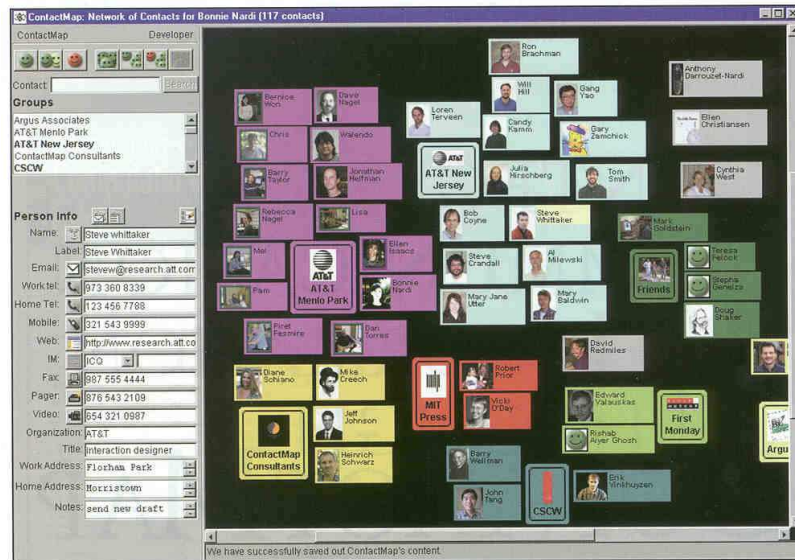


Fig. 24: Représentation '*Contact Map*' (source : [135])

Après avoir identifié les limites des métaphores actuelles de 'fichier' et de 'dossier' pour l'organisation des documents, ses auteurs ont conçu un nouveau type d'interface graphique qui organise l'environnement de travail de l'utilisateur sur base de ses contacts avec d'autres personnes. Cette approche repose sur la constatation que les réseaux de relations personnelles jouent un rôle fondamental dans l'accomplissement des tâches professionnelles. Les personnes sont représentées par leur photo dont positionnement dans l'espace d'affichage dépend des échanges qu'elles entretiennent entre elles. Des groupes de personnes tels que les amis ou les collègues sur un projet donné apparaissent donc clairement.

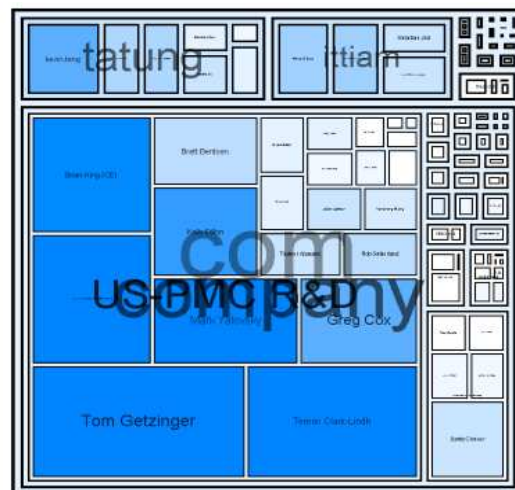


Fig. 25: Visualisation '*Correspondent Treemap*' (source: [151])

Dans une démarche similaire, Perer et Smith (2006) ont proposé le système '*Correspondent Treemap*'. Sur base de l'adresse de messagerie et de données disponibles sur la structure hiérarchique de l'organisation, l'appartenance des personnes de contact à différents niveaux de structuration de groupes est déterminée. Cette structure de données en arbre est visualisée

par un *treemap* qui associe le nombre de messages associés à chaque correspondant à la taille du rectangle qui lui fait référence. Le niveau de couleur (bleu) visualise le nombre de messages non lus. Plus le bleu est intense, plus il y a de messages non lus.

Certaines techniques de visualisation n'entrent pas directement dans une des quatre catégories de visualisations recensées par Viégas et ses collègues.

C'est notamment le cas des approches visant la **mise en relation graphique des messages électroniques et des tâches à accomplir**. Par exemple, Gwizdka (2002a, 2002b) a étudié la manière dont la messagerie électronique peut être utilisée pour gérer la question des tâches restant à accomplir (*pending tasks*). La représentation proposée (*TaskView*) se base sur une matrice dont l'axe horizontal est associé au temps et l'axe vertical à des tâches. Un message est représenté par une icône dans la cellule correspondant à sa date d'expédition et à la tâche à laquelle il se réfère. Cette représentation est très proche d'un diagramme de Gantt. L'originalité de la proposition réside plutôt dans l'établissement d'un lien entre tâches et courriels.

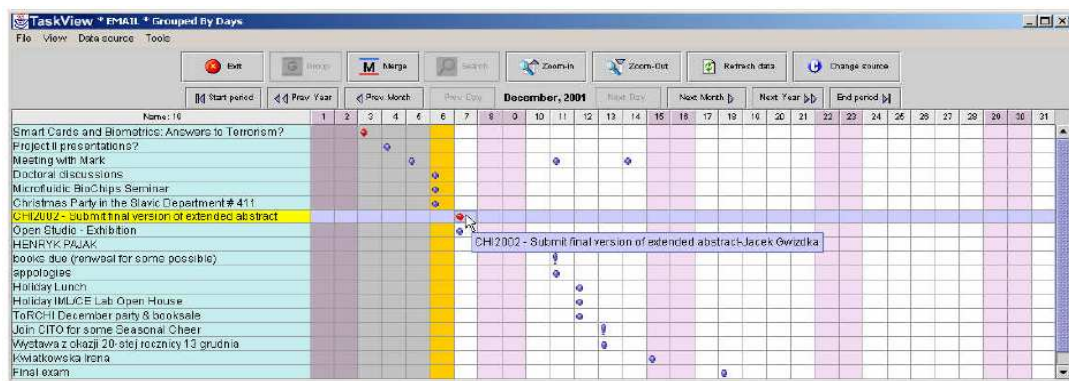


Fig. 26: Représentation 'TaskView' (source: [87])

Bellotti et ses collègues (2003) élargissent cette approche centrée sur les tâches en permettant non seulement la représentation de messages et de tâches mais aussi de fichiers et de liens (URL). Leur prototype, appelé '*TaskMaster*', est présenté à la Fig. 27.

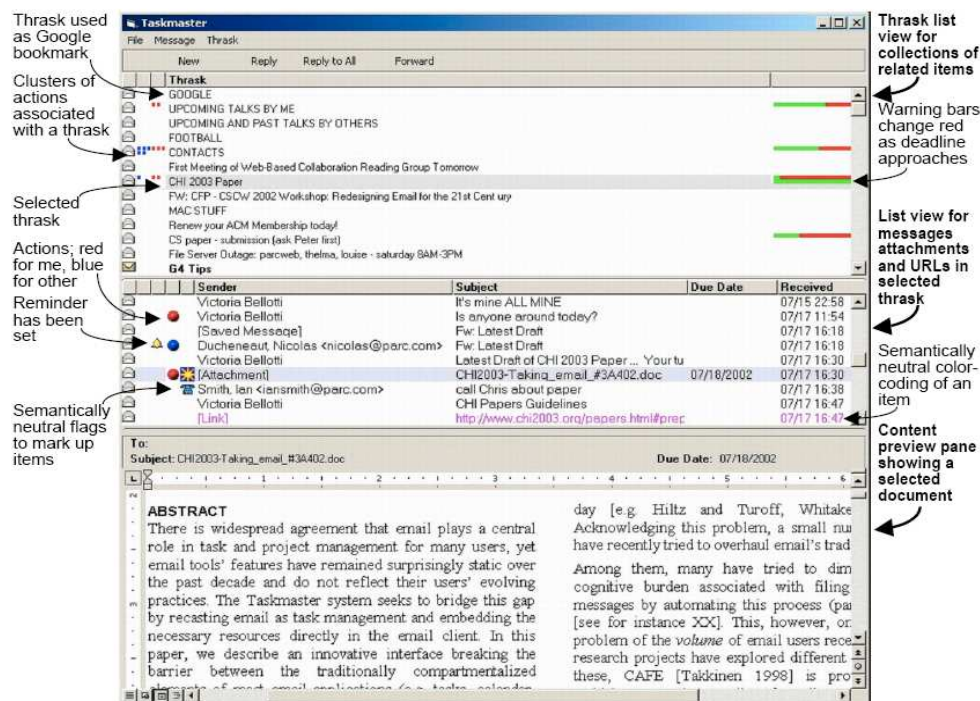


Fig. 27: Représentation 'TaskMaster' (source: [20])

Compte tenu de leur nombre, nous n'avons pas listé ici de manière exhaustive la multitude de travaux consacrés à la visualisation d'informations associées à des courriels. Néanmoins, nous pensons avoir proposé une vue d'ensemble représentative des grands courants de recherche portant sur cette thématique.

Nous concluons cette revue de différentes approches en mettant en évidence le fait qu'en ce qui concerne les données affichées, les différentes propositions se différencient par :

- le niveau de traitement des informations brutes,
 - Certaines représentations visualisent les messages eux-mêmes (p.ex. '*Thread Arcs*', '*VisuThread*') alors que d'autres montrent des informations calculées à partir d'une collection de messages (p.ex. '*Monthly Top Ten*', '*Themail*').
- la prise en compte d'éléments d'information extérieurs aux messages proprement dits,
 - Par exemple, la représentation '*Task View*' ajoute des informations sur les tâches auxquelles sont associés les messages.
- le caractère égo-centrique ou socio-centrique des données affichées,
 - Dans une représentation égo-centrique, on visualise les messages relatifs à une personne particulière (p.ex. '*Themail*', '*Soylent*') alors que dans une vue socio-centrique (p.ex. graphe SNA de la Fig. 17), les échanges au sein d'un groupe sont affichés sans favoriser un individu spécifique.
- l'entité sémantique mise en avant dans la visualisation.
 - La plupart des représentations utilisent le concept de message comme point de référence mais certaines approches se basent sur d'autres notions (p.ex. les personnes de contact dans '*Contact Map*').

1.2 Communications via chat et messagerie instantanée

Au cours des dernières années, les communications par messagerie instantanée (*Instant Messaging – IM*) et *chat* se sont répandues à une vitesse très importante. Elles se différencient de la messagerie électronique (communications asynchrones) par leur caractère de communications synchrones qui les destine particulièrement aux discussions informelles. Bien que nous traiterons cette question spécifique dans une section ultérieure, nous tenons à mentionner que la problématique de la visualisation de la présence des individus participant aux conversations ou observant celles-ci se pose de manière critique dans le contexte des communications synchrones.

La littérature contient de nombreux outils de visualisation de ce type de conversation. Dans cette section, nous en mentionnerons quelques-uns qui nous semblent particulièrement représentatifs des tendances observées.

La première catégorie de techniques regroupe les **visualisations destinées à supporter des conversations en cours**. Leur but principal est d'offrir des moyens de se rapprocher de conversations tenues en face-à-face. Elles utilisent donc divers artifices pour simuler des éléments considérés comme importants dans une discussion sans intermédiation électronique, tel que la dynamique des interactions, le sentiment de présence, la localisation spatiale ou la perception de l'environnement extérieur. Selon l'aspect de la conversation que l'on veut accentuer, des métaphores différentes seront utilisées. Leur raison d'être consiste donc dans l'ajout d'éléments aux simples messages textuels pour tenter d'enrichir la relation qu'ils supportent. Le concept de conscience de groupe (*awareness*) est très présent dans ces recherches.

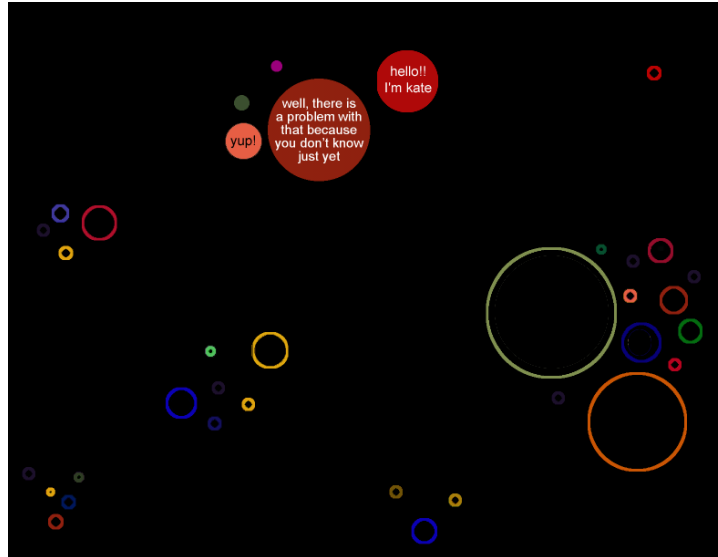


Fig. 28: Visualisation 'Chat Circles' (source: [54])

La représentation 'chatcircle' (Donath et al., 1999) est un bel exemple de ces travaux (cf. Fig. 28). Cette technique a été développée pour améliorer les utilitaires de 'chat'. Le principe en est assez simple. Chaque personne connectée au serveur de 'chat' apparaît comme un cercle de couleur différente. Quand un utilisateur poste un message, un cercle contenant le message apparaît en croissant progressivement à l'écran. Les cercles restent affichés pendant quelques secondes puis ils disparaissent à l'arrière plan. L'approche imite une conversation réelle où à chaque moment l'intérêt est porté sur les mots prononcés par la personne qui a parlé en dernier lieu. Au cours du temps, ces mots disparaissent et sont remplacés par de nouveaux. La représentation 'chatcircle' crée ainsi un rythme qui reflète celui d'une conversation réelle. L'identité des personnes est indiquée par la couleur du cercle, la position sur l'écran et par leur nom indiqué sur le cercle. La temporalité de la conversation n'est pas seulement représentée par l'apparition et la disparition des cercles mais aussi par la brillance du cercle, qui s'atténue avec le temps. Le système 'chatcircle' possède aussi une autre fonctionnalité intéressante. Toutes les personnes connectées au serveur 'chat' apparaissent sous la forme d'un cercle mais seuls les cercles situés dans une zone autour du cercle correspondant à l'utilisateur courant contiennent le texte des messages (cf. Fig. 28). Si celui-ci veut participer à une autre conversation, il doit se déplacer dans un autre endroit de l'écran. Cette fonctionnalité a été intégrée pour simuler le fait qu'une conversation en face-à-face est localisée et possède une portée spatiale limitée.

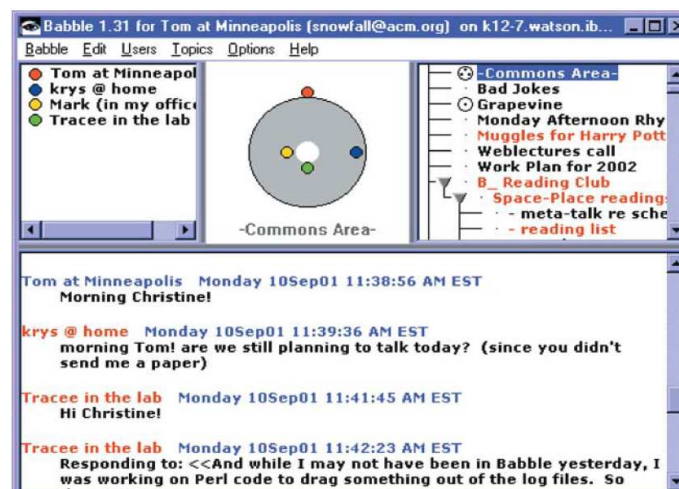


Fig. 29: Visualisation 'Babble' (source: [62])

Le système ‘*Babble*’ proposé par Erickson et ses collègues (1999, 2002) est un second exemple de support aux conversations par *chat* basé sur une visualisation originale. L’interface utilisateur de ‘*Babble*’ est présentée à la Fig. 29. D’un point de vue graphique, la zone la plus intéressante est celle située en haut au centre de l’interface que ses concepteurs désignent par le terme : ‘*Social Proxy*’. Son but est de fournir des informations sur la présence et l’activité des personnes participant à la conversation courante. Celle-ci est représentée par un disque gris. Les personnes connectées au système sont représentées par des points colorés. Si elles ne participent pas à la conversation, le point qui leur est associé est positionné en dehors du disque. Les points positionnés à l’intérieur du disque représentent les personnes actives dans la conversation. Ces points se déplacent vers le centre du disque quand les personnes parlent (‘*type*’) ou écoutent (‘*click and scroll*’) puis, dès que ces actions sont terminées, se dirigent progressivement vers la périphérie du disque. Cette technique de visualisation associe l’implication dans la conversation à un déplacement spatial qui peut rappeler celui souvent observé dans les rencontres face-à-face.

Le système ‘*Fugue*’ (Rosenberger et al., 2000) adopte une approche complètement différente (cf. Fig. 30). Il utilise la métaphore de la partition musicale pour mettre en évidence le rythme d’une conversation. Il a été conçu à partir de l’observation que les systèmes basés uniquement sur du texte ne permettaient pas de supporter le rythme des conversations. Or, celui-ci constitue une composante essentielle des conversations en face-à-face car il sert à contextualiser les mots. En effet, ses auteurs font remarquer qu’habituellement le rythme n’est pas redondant, en termes informationnels, avec la sémantique linguistique. Les mêmes mots prononcés avec un rythme et une intonation différente seront interprétés différemment.



Fig. 30: Visualisation ‘*Fugue*’ (source: [171])

‘*Fugue*’ représente les messages postés comme une partition de musique (cf. Fig. 30). Le temps est associé à l’axe horizontal et les différents intervenants sont associés à l’axe vertical. Comme dans une partition musicale, si deux événements se produisent en même temps, ils sont positionnés sur la même verticale. ‘*Fugue*’ enregistre les caractères composant les messages et les visualise en temps réel pour représenter le rythme de la conversation. La zone horizontale associée à chacun des participants est un indicateur de sa présence sociale (*social presence*). Elle apparaît lors de la connexion et disparaît à la déconnexion. La taille des caractères associés à chaque utilisateur est liée à son degré d’activité récent. Elle diminue à mesure qu’augmente le temps écoulé depuis la dernière intervention de la personne concernée. Cette fonctionnalité participe également à traduire la dynamique de la conversation en attirant l’attention sur les personnes qui ‘prennent la parole’ (c.-à-d. qui postent un nouveau message).

L’enrichissement de la conversation peut également être opéré en l’insérant dans un monde virtuel qui rappelle des composantes de la vie réelle. Un exemple classique consiste à associer les personnes à des avatars placés dans un monde virtuel 3D (cf. p.ex. ‘*Palace*’, 2005 ; ‘*V-Chat*’ décrit dans Smith et al., 2000). Ces systèmes essaient de proposer un ersatz virtuel de l’environnement spatial dans lequel les conversations réelles ont lieu. Le système ‘*Comic Chat*’ (Kurlander et al., 1996) qui représente les participants à la conversation sous la forme

de personnages dans une bande dessinée (cf. Fig. 32) peut aussi être associé à ce type de démarche.



Fig. 31: Système 'Palace' (source : [147])

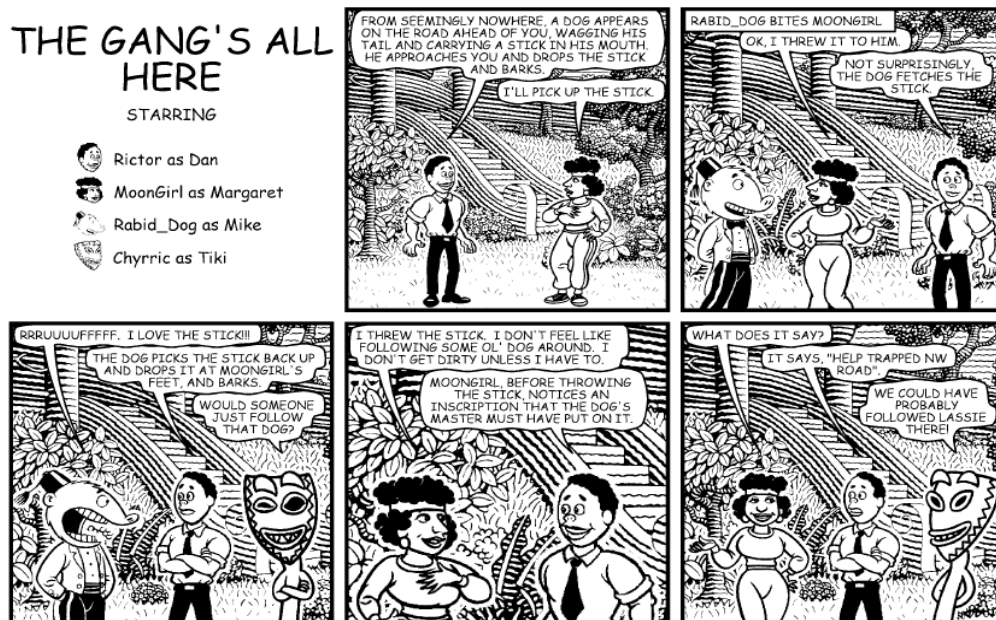


Fig. 32: Système 'Comic Chat' (source : [108])

Le second type de visualisations associées au *chat* et à la messagerie instantanée rassemble les approches ayant pour vocation de **faire apparaître la structure des communications au sein de groupes**. Leur but réside donc dans une meilleure compréhension des interactions sociales. Par exemple, il s'agit de faire apparaître les leaders ou les fauteurs de trouble.

En sus des classiques représentations sous forme de graphes, on rencontre des propositions plus originales telles que celles issues du projet *Loom* (Donath et al., 1999 ; Social Media Group, 2006). Une première vue représente les messages postés sous la forme de points lumineux placés dans une grille bi-dimensionnelle dont les axes sont associés au temps et aux participants. Une conversation est affichée sous la forme d'une ligne qui connecte les points relatifs aux messages qui la compose (cf. Fig. 33a). Selon ses concepteurs, cette technique de visualisation fournit une image de l'atmosphère du groupe de discussion. Elle indique, par exemple, si les débats sont passionnés et si les conversations passent rapidement d'un participant à un autre.

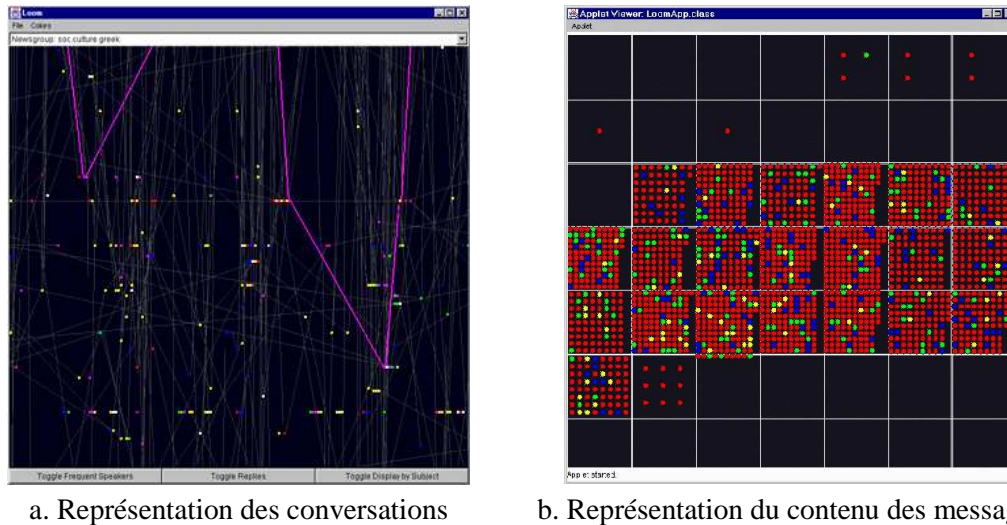


Fig. 33: Visualisation ‘Loom’ (source: [189])

Une seconde technique issue du projet *Loom* (cf. Fig. 33b) repose sur une catégorisation automatique des messages postés dans des catégories telles que ‘fâché’, ‘calme’, ‘informationnel’... A chaque catégorie est associée une couleur. Les messages sont visualisés par des points colorés placés sur une grille représentant un calendrier. Cette vue est sensée donner une impression sur l’humeur au sein d’un groupe de discussion.

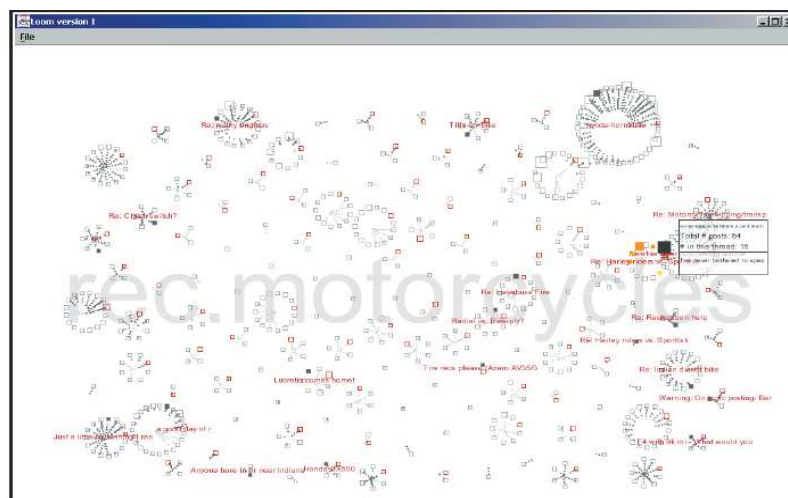


Fig. 34: Visualisation ‘Loom’ (source: [53])

Une troisième visualisation du projet ‘Loom’ (Donath, 2002) présente les messages en plaçant dans un cercle ceux qui sont reliés entre eux (cf. Fig. 34). L’idée est de faire allusion visuellement à des espaces dans lesquels des conversations intenses et animées se produisent. Les messages sans relation avec d’autres apparaissent comme des points isolés, ce qui illustre l’absence d’interactions sociales à ce niveau.

Les techniques de visualisation ont aussi été utilisées pour rendre persistentes les conversations basées sur des technologies de communication synchrone qui sont, par nature, relativement éphémères. A cette fin, les propositions portent sur la **visualisation de l’historique des communications**.

Par exemple, le système ‘Babble’ (Erikson et al., 2002) offre un moyen de représenter l’historique des communications appelé ‘Timeline’ (cf. Fig. 35). Il repose sur l’utilisation très classique d’une ligne du temps. Chaque utilisateur est associé à une ligne horizontale qui

indique quand il est connecté. La ligne est colorée quand l'utilisateur associé prend part à la conversation visualisée, sinon elle est grise. Cette représentation permet, entre autres, d'identifier certains comportements sociaux (p. ex. voir les utilisateurs qui se connectent pour voir ce qui se passe dans la conversation puis la quitte rapidement).

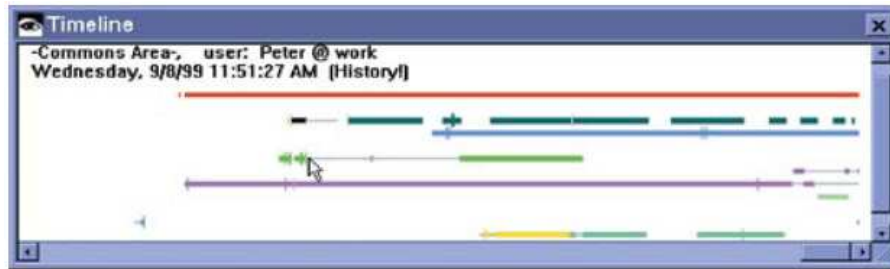


Fig. 35: Visualisation 'Babble Timeline' (source: [62])

La visualisation 'CrystalChat' (Tat et Carpendale, 2006) se concentre plutôt sur la visualisation de l'historique des communications individuelles. Son but est de faire émerger des motifs (*patterns*) représentatifs de conversations avec les interlocuteurs de la personne de référence. La visualisation est conçue en combinant la structure du réseau social et l'aspect temporel des communications dans une représentation 3-D.

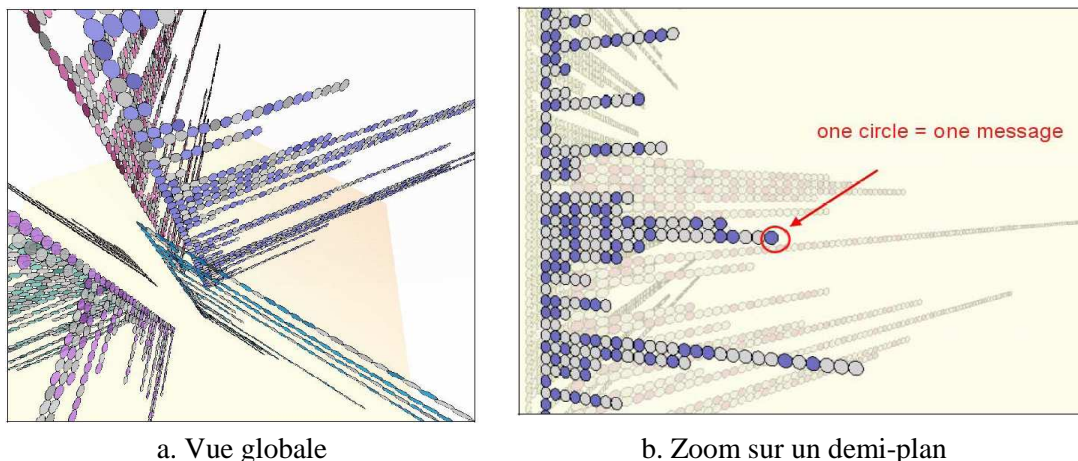


Fig. 36: Visualisation 'CrystalChat' (source: [196])

La visualisation se présente sous la forme d'un ensemble de demi-plans dont l'intersection est associée à l'utilisateur de référence (cf. Fig. 36a). Chaque demi-plan est associé à un de ses interlocuteurs (cf. Fig. 36b). L'axe vertical est associé au temps. Les messages entre l'utilisateur de référence et une personne donnée P sont représentés sous la forme de points placés sur le demi-plan P correspondant. Chaque personne est associée à une couleur. Chaque alignement horizontal de points dans le demi-plan P correspond à une conversation. L'utilisateur peut interagir dynamiquement avec la représentation tri-dimensionnelle pour visualiser un sous-ensemble de données particulier.

Le système 'Fugue' (Rosenberger et al., 2000) présenté précédemment offre également une fonctionnalité de visualisation de l'historique des communications (cf. Fig. 37). Très classiquement, le temps est associé à l'axe horizontal. L'interface se compose de deux zones principales : la zone inférieure qui représente les périodes d'activité par des lignes horizontales continues et offre une vue d'ensemble et la zone supérieure qui visualise le détail de la communication pendant une sous-période. Le couplage entre les deux zones est réalisé par le biais d'un curseur situé dans la zone inférieure. Une originalité du système réside dans la possibilité d'annoter l'historique et de diffuser ces annotations par courriel. Les notes apparaissent sous la forme de points lumineux dans la partie supérieure de l'interface.

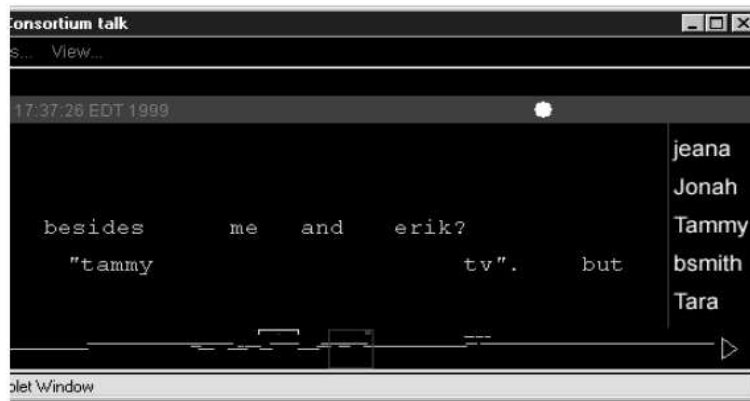


Fig. 37: Visualisation d'historique dans 'Fugue' (source : [171])

En conclusion, les techniques de visualisation associées aux systèmes de communication synchrone tels que le *chat* ou la messagerie instantanée présentent des différences importantes par rapport à celles focalisées sur la messagerie électronique. Ces différences reposent en grande partie sur la distinction entre communication synchrone et asynchrone. Lorsqu'il est question de concevoir des visualisations associées à des communications synchrones, des questions nouvelles se posent par rapport aux échanges basés sur la messagerie électronique (*e-mail*) en terme :

- de support en temps réel de la communication en cours (cf. p.ex. 'Fugue'),
- d'enrichissement du cadre de la communication pour se rapprocher des communications en face-à-face (cf. p.ex. 'Babble', 'Chat Circles', 'Palace'),
- de transformation de communications éphémères en données persistentes (cf. p.ex. 'CrystalChat'),
- d'accès à des données de type socio-centriques (en général, il est beaucoup plus aisé et accepté de visualiser les messages de tout un groupe que dans le cas de la messagerie électronique),

1.3 Groupes de discussion, forums

Nous nous focalisons dans cette section sur les technologies de discussion de groupe telles que groupes de discussion (*newsgroup*), forums ou encore tableaux de messages (*message board*). Comme la messagerie électronique, ces média de communication sont (plutôt) asynchrones et supportent la persistance des données. Par contre, ils s'en différencient par les contextes dans lesquels ils sont utilisés :

- regroupement thématique,
- contexte plus informel que la messagerie électronique,
- visibilité des messages plus large que la messagerie électronique.

Les techniques de visualisation associées peuvent être regroupées dans des catégories similaires à celles associées aux autres média. On retrouve des représentations graphiques focalisées sur la structure des communications au sein de groupes de personnes, sur l'aspect historique, ...

Nous présentons ci-dessous quelques exemples représentatifs des travaux dans ce domaine.

Newsgroup Crowds (Viégas et Smith, 2004) est une technique de visualisation associée à l'espace de discussion *Usenet* (*newsgroup*). Elle a pour but de montrer l'activité de participants dans un groupe de discussion au cours d'une période de temps donnée. La visualisation (cf. Fig. 38 a) représente les personnes par des cercles dont le positionnement indique le nombre de jours pendant lesquels une personne particulière a été active pendant le mois sélectionné (axe vertical) et le nombre moyen de messages postés (*posts*) par

conversation (*thread*) (axe horizontal). La couleur des cercles indique si une personne a été active récemment. En cliquant sur un cercle, on peut visualiser une fiche descriptive de l'activité de la personne dans une fenêtre semi-transparente (cf. Fig. 38 b). *Newsgroup Crowds* se classe dans la catégorie des **représentations de la structure des communications au sein d'un groupe**. Selon ses concepteurs, elle permet de mettre en évidence différents types d'intervenants dans les groupes de discussion (p.ex. nouveaux venus, personnes aimant débattre).

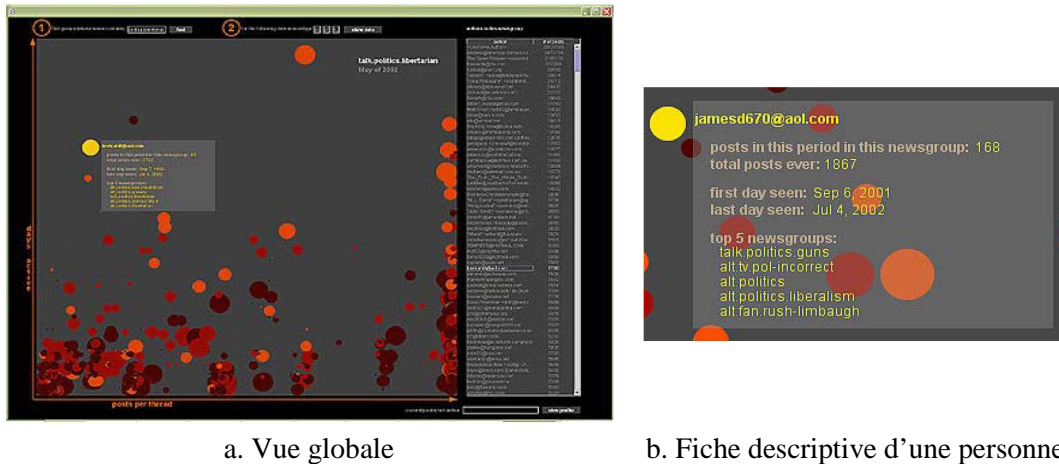


Fig. 38: Visualisation 'Newsgroup Crowds' (source: [206])

PeopleGarden (Donath, 2002) est une autre approche. Elle utilise la métaphore du jardin pour visualiser la participation à un tableau de message (*message board*). Chaque participant est représenté par une fleur dont la longueur de la tige indique le temps passé par la personne dans le système et le nombre de pétales le nombre de messages postés. Les messages initiaux sont affichés en rouge et les réponses en bleu. Dans son analyse de cette technique, Donath indique cette métaphore a du sens car elle représente un groupe actif comme un beau jardin rempli de fleurs (cf. Fig. 39, partie droite) alors qu'un groupe peu actif ressemble à un jardin négligé (cf. Fig. 39, partie gauche).

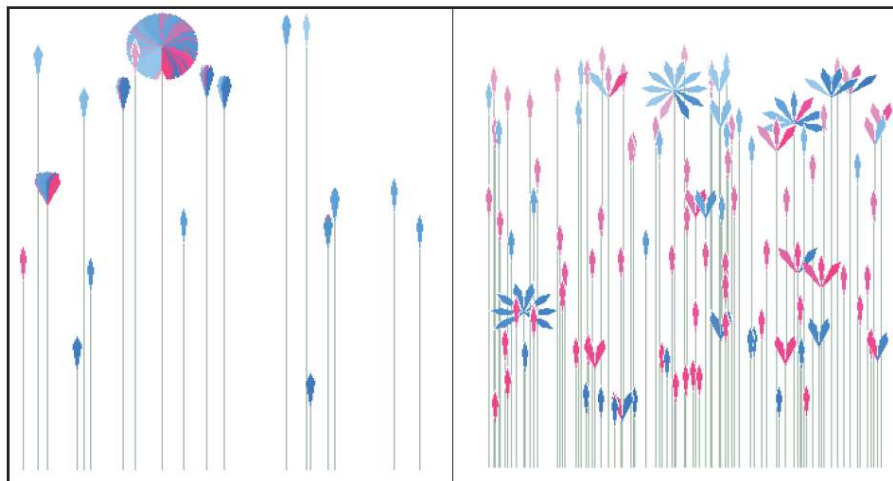


Fig. 39: Visualisation 'PeopleGarden' (source: [53])

Néanmoins, Donath identifie un risque avec ce type de visualisations : *'le problème c'est la surcharge sémantique de la représentation symbolique compte tenu de la minceur de l'information à visualiser. Si une visualisation évoque une signification qui va au-delà de la correspondance directe avec les données, il doit y avoir de l'information dans ces données qui guident le choix et la forme de la visualisation.'* Par exemple, dans *PeopleGarden*, il

existe une correspondance directe entre les comportements de dépôt de message (*posting patterns*) et la croissance et la forme des fleurs mais la métaphore de la fleur et du jardin possède également sa propre signification métaphorique qui doit être considérée quand cette visualisation est utilisée.

Authorlines (Viégas et Smith, 2004) est une technique de visualisation portant sur les groupes de discussion de *Usenet*. Elle se focalise sur l'historique de l'activité d'une personne P dans les différents groupes de discussion auxquels elle participe au cours d'une année. La représentation se présente comme suit (cf. Fig. 40).

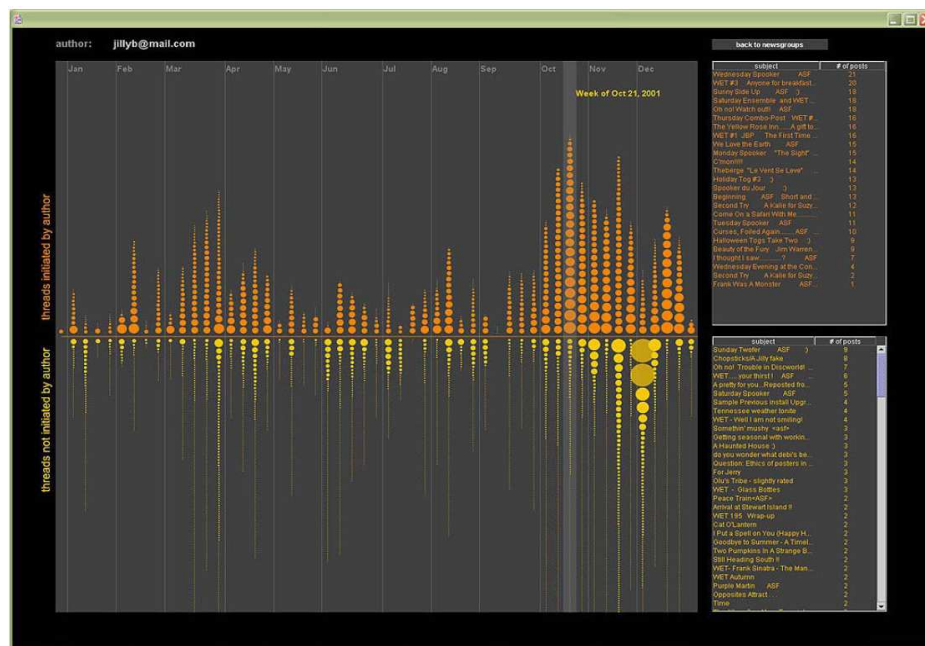


Fig. 40: Visualisation 'Authorlines' (source: [206])

L'axe du temps est associé à l'axe horizontal. L'activité hebdomadaire de la personne P est représentée par un empilement vertical de cercles qui sont associées aux conversations auxquelles elle a participé. La taille des cercles est liée au nombre de messages postés par P dans la conversation concernée. Les conversations sont positionnées au-dessus ou en-dessous de la ligne du temps selon qu'elles aient été initiées par P ou non. *Authorlines* se range donc dans la catégorie des **visualisations à caractère temporel**, au côté d'approches telles que *TheMail* (cf. Fig. 22) ou *Monthly TopTen* (cf. Fig. 21). Ces représentations font apparaître la manière dont les communications se distribuent au cours du temps.

Pour conclure, quand il est question de visualiser des données associées à des groupes de discussion, une préoccupation importante concerne l'analyse sociale du groupe et l'utilisation de méthodes graphiques pour qualifier la structure du groupe, pour identifier des sous-groupes ou encore pour déceler des comportements individuels particuliers. Néanmoins, certains auteurs nous mettent en garde contre l'utilisation de techniques visuelles qui biaisent l'interprétation de la réalité sociale des communications.

2. Visualisations d'actions sur des ressources partagées

Les opérations réalisées sur des ressources partagées constituent, à coté des communications, une autre composante essentielle d'un environnement collaboratif. Il est donc important que les participants en aient conscience. Les techniques de visualisation sont aussi utilisées dans cette optique.

Compte tenu de l'objet de notre réflexion globale, nous limiterons notre étude de la littérature aux questions associées aux ressources partagées électroniques. Nous tenons néanmoins à

mentionner que nous sommes parfaitement conscients qu'il ne s'agit que d'un sous-ensemble des ressources partagées dans un contexte collaboratif global.

Par ailleurs, il faut remarquer que les communications électroniques reposent aussi sur l'accès à des ressources partagées électroniques (p.ex. serveur de messagerie électronique). Dès lors, il est nécessaire de préciser notre définition de ce concept. Nous considérerons **qu'une ressource électronique partagée est un objet informatique auquel plusieurs personnes ont accès en lecture ou en écriture dans le but de le créer puis, éventuellement, de le faire évoluer pour obtenir une version acceptée par tous ou répondant à une spécification donnée.**

Les ressources partagées auxquelles nous nous référerons peuvent être regroupées en plusieurs catégories. La première rassemble les ressources électroniques partagées les plus fréquentes : à savoir les documents. Dans notre acceptation, le terme 'document' inclut non seulement les documents classiques issus de logiciels de bureautiques (p.ex. traitement de texte, tableur...) mais aussi des pages web (cf. Systèmes de gestion de contenu – *Content Management Systems*), des documents inclus dans des environnements collaboratifs (p.ex. Documents internes de la suite Lotus Notes) ou des programmes informatiques (cf. systèmes de développement collaboratif de logiciels). Les documents sont donc essentiellement constitués de représentations abstraites, souvent majoritairement textuelles. Une seconde catégorie importante regroupe les modèles numériques partagés utilisés dans des domaines tels que l'ingénierie mécanique, la médecine ou l'architecture. Nous les classons dans une catégorie spécifique car les interactions se produisent en majorité en manipulant des représentations graphiques (2-D ou 3-D) qui montrent des propriétés physiques d'objets. Cette classification peut donc être rapprochée de la distinction habituellement adoptée entre 'visualisation d'information' et 'visualisation scientifique'.

En ce qui concerne la visualisation des opérations réalisées sur des ressources partagées, on peut distinguer deux grandes approches. La première consiste à représenter le contenu informationnel de la ressource et d'indiquer graphiquement les différences par rapport à une autre version. Nous appellerons cette approche : 'visualisation différentielle du contenu'. La seconde approche se limite à visualiser des métadonnées (p.ex. date de dernière modification, taille, auteur...) relatives aux ressources partagées. Par exemple, les versions d'un document sont associées à des glyphes positionnés sur une ligne du temps. Nous la désignerons par le terme : 'visualisation de méta-données'.

Nous avons encore identifié un troisième axe d'analyse des techniques de visualisation d'opérations sur des ressources partagées : l'élément mis au centre de la représentation. Deux grandes tendances peuvent être rencontrées : la focalisation sur les ressources partagées (*resource-centric view*) ou sur les personnes qui réalisent des opérations sur elles (*person-centric view*).

En conclusion, les techniques de visualisation d'opérations sur des ressources partagées se différencient donc notamment par :

- la nature des ressources partagées concernées : documents plutôt textuels ou modèles numériques,
- les informations représentées : différence entre le contenu ou méta-données,
- l'élément contral mis en évidence : les ressources ou les personnes qui interagissent avec elles.

Nous présentons dans la suite de cette section quelques travaux représentatifs des approches rencontrées dans la littérature.

La fonctionnalité '*Track Changes*' du traitement de texte *Microsoft Word* (cf. Fig. 41) constitue un exemple très connu de visualisation différentielle du contenu d'un document. Il se base sur la visualisation combinée de plusieurs versions d'un texte différenciées essentiellement par la couleur de la police de caractère utilisée.

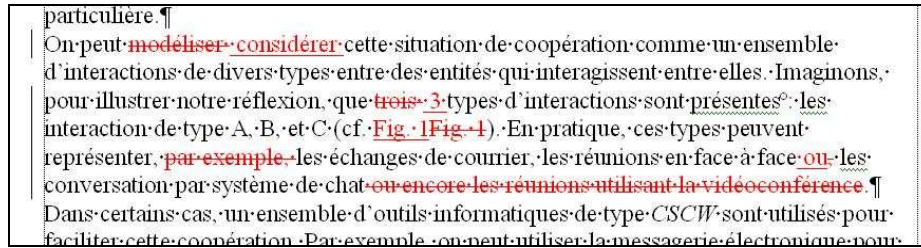


Fig. 41: Système 'Track changes' de Microsoft Word

Dans le domaine des modèles numériques, une approche conceptuellement très proche consiste à visualiser l'état du modèle à différents moments pour montrer son évolution. C'est par exemple l'approche retenue par le prototype 'Archichronos' (Riendeau, 2006) pour représenter l'évolution d'un bâtiment au cours de son processus de conception (cf. Fig. 42). Les modèles 3-D du bâtiments sont superposés et un codage de couleur indique si les éléments ont été modifiés, ajoutés, supprimés ou sont restés inchangés d'une version à une autre.

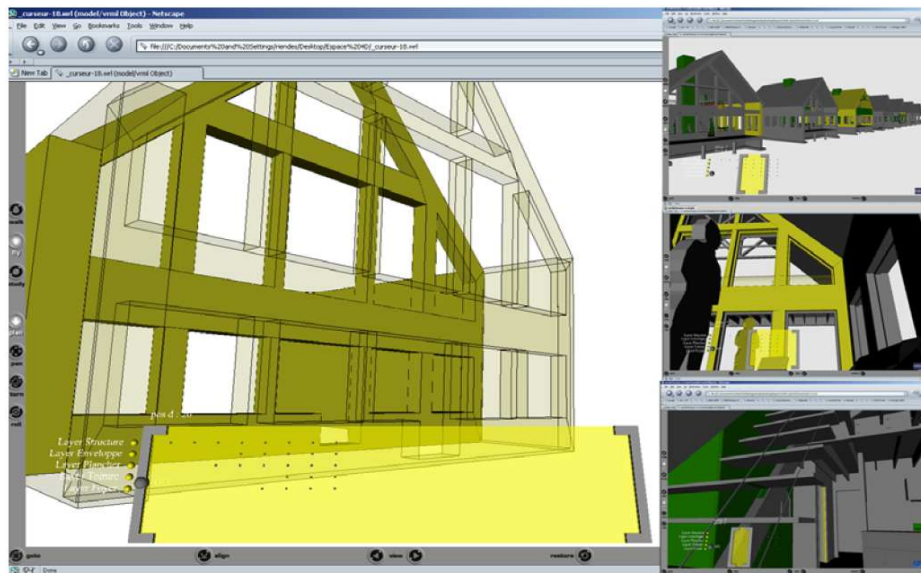


Fig. 42: Prototypé Archichronos (source: [160])

Les outils d'annotation de ressources partagées appartiennent également à cette catégorie. Les annotations constituent en effet des ajouts ou des retraits de contenu informationnel des ressources. A titre d'exemple de ce type d'outils, on peut citer les tableaux blancs partagés ('*shared whiteboard*') qui consistent à permettre à plusieurs utilisateurs d'écrire sur un espace commun de manière synchrone.

Plusieurs travaux de recherche ont étudié la problématique de la représentation graphique des opérations effectuées sur des documents partagés en visualisant plutôt des méta-données associées.

La technique la plus classique consiste à représenter les différentes opérations dans une liste séquentielle en faisant appel, explicitement ou implicitement, à la notion de ligne du temps.

C'est celle qui a été proposée par St Ger et Engeli (1998) qui ont développé des nouvelles interfaces visuelles visant à supporter les activités au sein de projets conduits par des entreprises virtuelles. Une de leurs idées clés consiste à créer des méta-vues sur les données. Une méta-vue est une '*représentation qui montre en premier lieu la structure des relations entre les documents plutôt que leur contenu.*' Parmi les diverses méta-vues qu'ils proposent, nous mentionnerons celle appelée '*TimeBand*' basée sur une représentation linéaire du temps (cf. Fig. 43). Elle permet de visualiser les documents et les acteurs du projet et fournit une bonne vue d'ensemble du travail en cours. Selon ces chercheurs, '*l'organisation des*

documents sur base temporelle est très utile pour la tâche de recherche de documents spécifiques car les utilisateurs peuvent souvent se souvenir de quand un document a été écrit ou accédé ou à tout le moins ils peuvent reconnaître la séquence des documents.’ Outre la vue globale des documents, il est possible d’accéder à certaines méta-données y relatives (ex. titre, auteur, mots-clés...), en pointant sur le carré représentant un document spécifique. Des couleurs différentes peuvent être associées aux carrés représentant les documents selon différents axes de différenciation : type de document, statut du document...

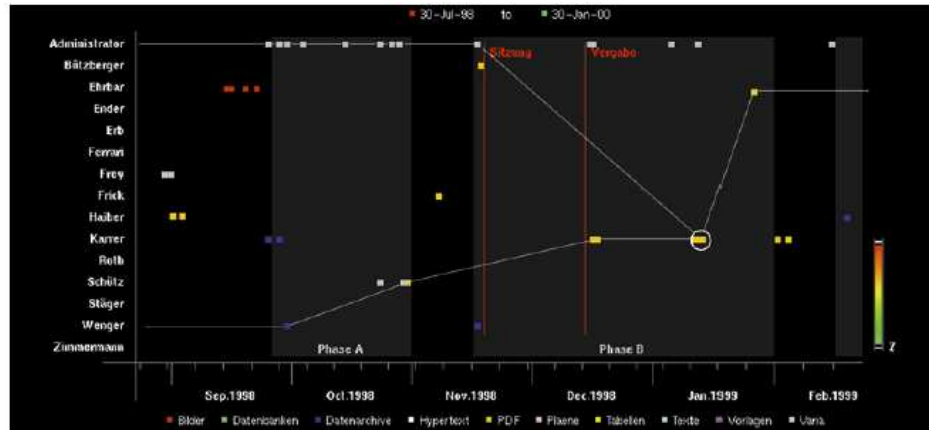


Fig. 43: Méta-vue 'TimeBand' (source : [194])

Dans le système HyperNavi, Noirhomme et Serpe (Noirhomme-Fraiture et Serpe, 1998) proposent une autre approche qui consiste à visualiser des travaux parlementaires en y insérant des liens hypertextes vers des ressources permettant de rendre compte de leur construction.

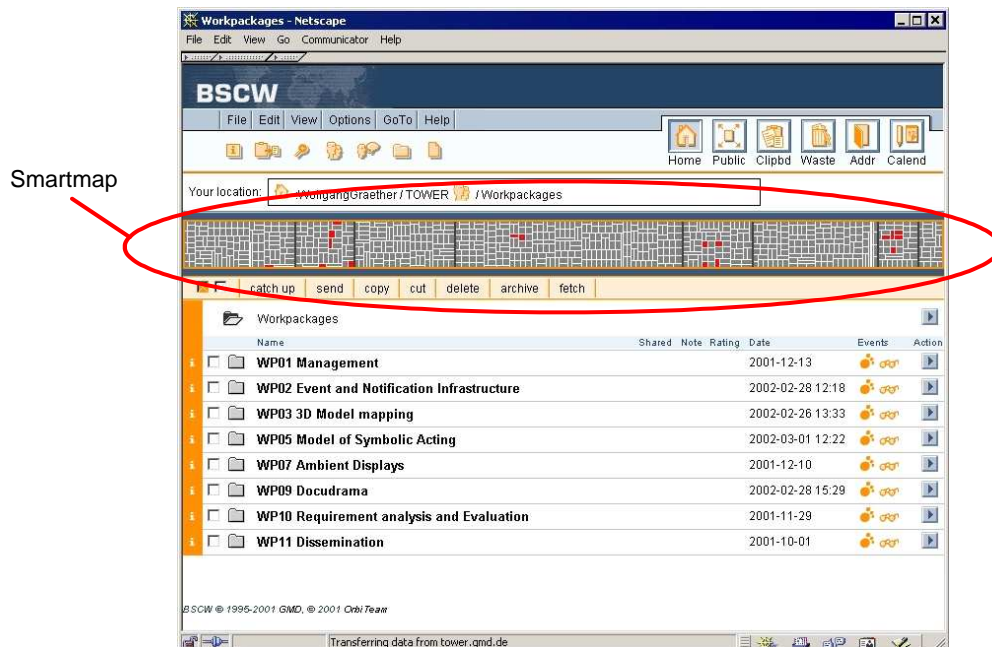


Fig. 44: Smartmaps (source : [158])

Prinz et son équipe (Prinz et al., 2004 ; Gräter et Prinz, 2005) ont proposé une technique plus élaborée. Ils ont utilisé, sous le nom de *smartmaps*, des *treemaps* dans le contexte d’environnements collaboratifs. Parmi diverses fonctionnalités, celle qui nous intéresse ici se révèle être le codage par couleur des rectangles associés aux ressources qui permet de connaître les opérations passées ou présentes réalisées sur celles-ci. Les *smartmaps* ont

notamment été intégrés à la plate-forme de coopération BSCW (cf. Fig. 44). Il faut noter que la représentation par *treemap* nécessite que l'information à visualiser soit structurée sous forme arborescente.

Le système '*LifeSource*' (Gilbert et Karahalios, 2006) est un autre exemple, issu du domaine du suivi de développement logiciel. L'exemple choisi est relatif à un outil CVS (*Concurrent Versioning System*) de développement en langage Java largement utilisé dans le monde du logiciel libre. Deux techniques de visualisation sont proposées dans ce cadre : '*CodeConnections*' qui est centrée sur la notion de fichier (*file-centric*) et '*CodeSaw*' qui est focalisée sur les auteurs du code (*author-centric*).

L'interface utilisateur de '*CodeConnections*' (cf. Fig. 45) se divise en deux parties. La zone supérieure montre la structure du projet à un moment donné et la zone inférieure regroupe les moyens offerts à l'utilisateur pour contrôler la navigation dans la représentation.

La visualisation de la structure du projet se base sur un graphe. Les nœuds sont associés aux fichiers de code et sont représentés par des carrés dont la taille est associée à la taille et de l'importance du fichier. Les arcs visualisent les relations d'appartenance à un même '*package*' ou celles d'héritage. Si on clique sur le nom d'un développeur, la contribution de celui-ci dans le codage de chaque fichier est visualisée en superposant un carré coloré au carré représentant le fichier (cf. Fig. 45, partie gauche). L'interface offre aussi la fonctionnalité classique de vue d'ensemble (cf. Fig. 45, zone centrale inférieure) couplée à une vue détaillée mais partielle des informations représentées. Notons enfin que *CodeConnections* permet également de visualiser la dynamique des travaux de développement en animant le graphe.

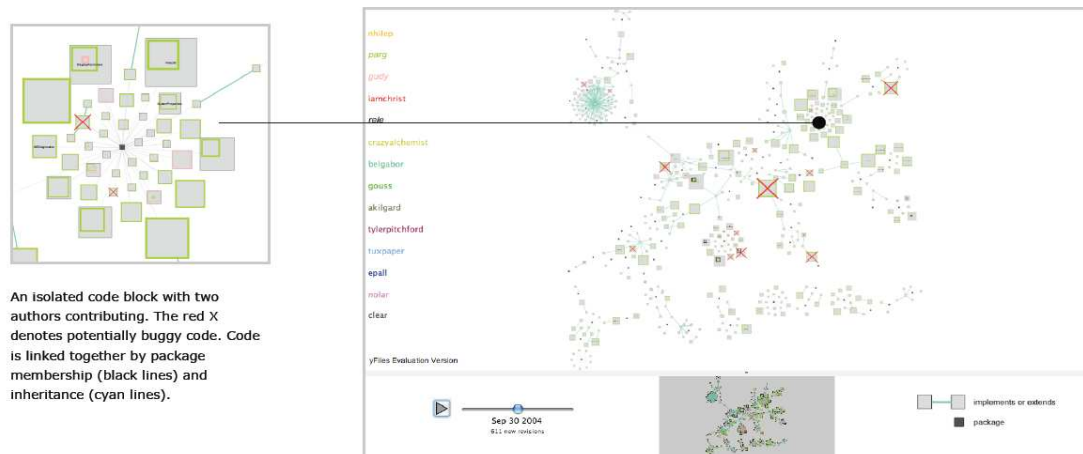


Fig. 45: Visualisation '*CodeConnections*' (source: [70])

Selon ses concepteurs, *CodeConnections* peut aider les développeurs à bien comprendre comment les différentes pièces du code s'intègrent les unes aux autres. Il peut aussi révéler des tendances historiques dans le développement du code en identifiant, par exemple, des fichiers qui disparaissent.

La seconde technique, appelée *CodeSaw*, a pour vocation de montrer les contributions des développeurs au cours d'une année (cf. Fig. 46).

Chaque personne est associée à une ligne horizontale L_i , qui fait office de ligne du temps. Les contributions de l'auteur A au code sont représentées par des triangles situés au-dessus de la ligne L_A et ses contributions à la liste de diffusion (*mailing list*) par des triangles situés sous cette ligne. Les développeurs sont classés selon l'importance combinée de leur contribution au code et à la liste de diffusion.

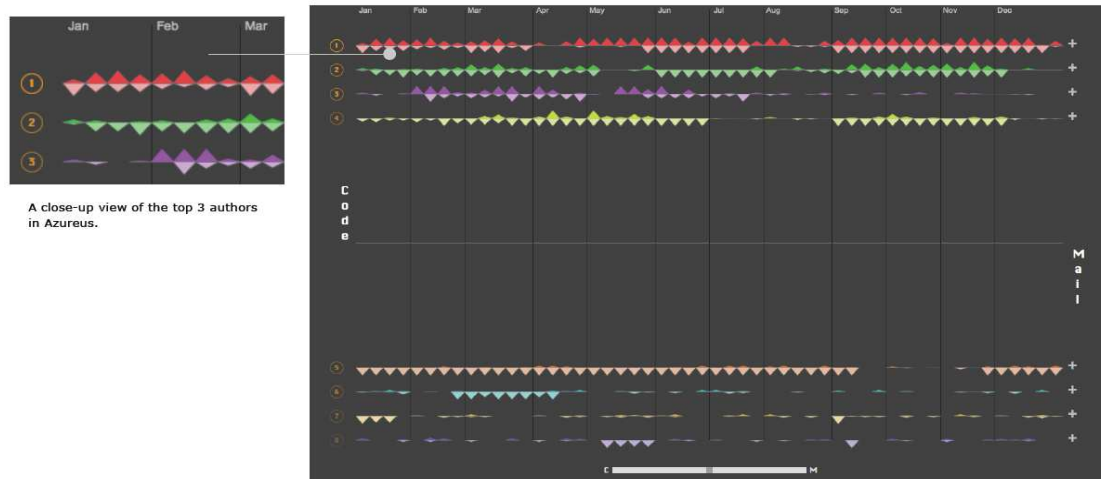


Fig. 46: Visualisation ‘CodeSaw’ (source: [70])

Certains travaux combinent la représentation basée sur les méta-données avec une certaine visualisation du contenu différentiel.

C’est, par exemple, l’approche adoptée par la technologie ‘wiki’ qui permet l’édition des pages d’un site web par ses visiteurs (cf. Fig. 47). Certaines méta-données sont combinées avec des informations succinctes sur la différence entre les versions successives.

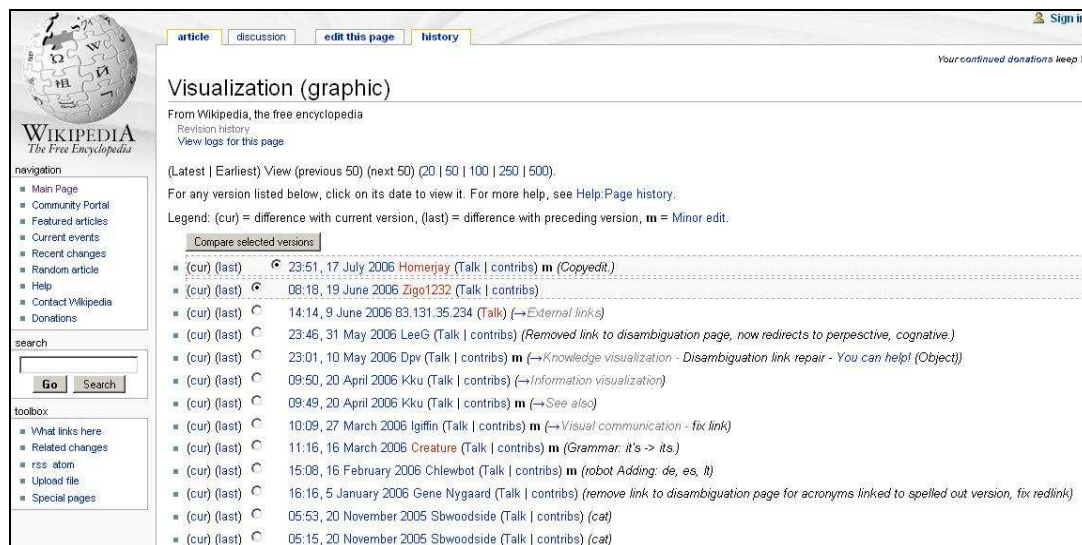


Fig. 47: Représentation des opérations sur un document dans Wikipedia (source: [220])

Viégas et ses collègues (2004) ont étudié la représentation des opérations effectuées sur un document dans ce même contexte. Leur système, appelé ‘History Flow’, a pour but de rendre immédiatement visibles les grandes tendances historiques concernant les révisions des pages web.

Le principe constructif de la visualisation est illustré par la Fig. 48 a. Supposons que trois personnes modifient un document (cf. Fig. 48 a (A)). Chaque version du document est représentée par une ligne composée de segments colorés indiquant l’auteur de la modification et la longueur de celle-ci. Cette représentation est intéressante mais elle ne permet pas de connaître la relation entre les versions. L’idée fondamentale à la base de la technique ‘History Flow’ est de relier visuellement les sections de texte qui restent identiques entre deux versions consécutives. Pour ce faire, des zones ombrées sont dessinées entre les segments de droite consécutifs qui se correspondent (cf. Fig. 48 a (B)). Les parties de texte qui n’ont pas de correspondance dans la version précédente ou suivante ne sont pas connectées. L’utilisateur voit donc clairement les insertions et les suppressions dans le texte. Deux versions du système

‘History Flow’ sont proposées : l’une dispose les droites verticales de manière équidistante (cf. Fig. 48 a (B)) et l’autre met en évidence la dynamique des modifications en ajoutant une ligne du temps (implicite) à l’axe horizontal (cf. Fig. 48 a (C)). La technique ‘History Flow’ a été utilisée pour étudier le rythme de collaboration entre les auteurs du site ‘wikipedia’ (cf. Fig. 48 b).

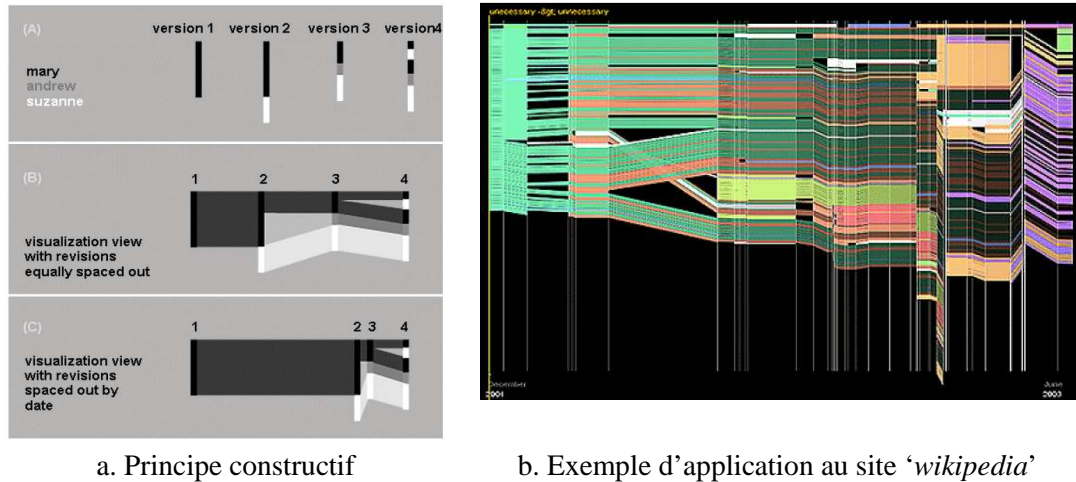


Fig. 48: Système ‘History Flow’ (source: [207])

En conclusion, nous avons identifié trois axes de différenciation des techniques de visualisation d’opérations sur des ressources partagées : la nature des ressources (documents ou modèles numériques), le type d’informations visualisées (différences de contenu ou méta-données) et l’élément central choisi comme point de focalisation de la représentation (ressources ou personnes). Ces critères peuvent être utilisés pour classifier les techniques rencontrées dans la littérature.

Il faut noter que les techniques décrites précédemment sont mentionnées à titre illustratif des différentes approches et ne peuvent être considérées comme un état de l’art exhaustif.

3. Visualisations de la présence dans un environnement partagé

Avant toute autre considération, nous devons préciser notre définition d’environnement partagé dans le cadre de la visualisation de la présence de plusieurs personnes. Dans notre acceptation, il s’agit de **systèmes socio-techniques servant d’environnements de référence à un groupe de personnes en situation de collaboration**. Cette définition inclut les environnements informatiques partagés formels tels que les plates-formes informatiques de coopération mais aussi les systèmes formés par les utilisateurs d’un outil de messagerie instantanée ou un répertoire de fichiers partagés. Nous imposerons comme limite qu’ils reposent sur l’utilisation de technologies informatiques.

Le concept de présence dans un environnement partagé est indissociable de celui de conscience de groupe. Quel que soit le type de conscience de groupe concernée (*informal, social, structural, workspace awareness*, cf. p. 21), il est toujours fait référence à la communication d’informations relatives à la notion de présence des autres personnes : qui elles sont, comment elles se comportent, avec quoi elles interagissent... Compte tenu de l’importance de cet élément dans le travail de groupe assisté par ordinateur, de nombreux travaux de recherche ont porté sur les moyens de le représenter graphiquement. C’est la raison pour laquelle nous avons introduit la catégorie : ‘*Visualisations de la présence dans un environnement partagé*’ bien qu’elle ne soit pas orthogonale avec les catégories ‘*Visualisations de communications*’ et ‘*Visualisations d’actions sur des ressources partagées*’.

L'objectif principal dans cette thématique de recherche consiste à enrichir la représentation électronique des personnes avec des données sémantiquement correctes, socialement acceptables et perceptibles sans biais. A la différence d'autres types de données, le caractère hautement personnalisé des données de présence ajoute des difficultés supplémentaires en termes de perception psychologique, d'acceptation sociale ainsi que de protection de la vie privée. En effet, les connotations subjectives associées à ce type de représentation graphique peuvent s'avérer déterminantes dans la perception et donc l'acceptation des approches proposées.

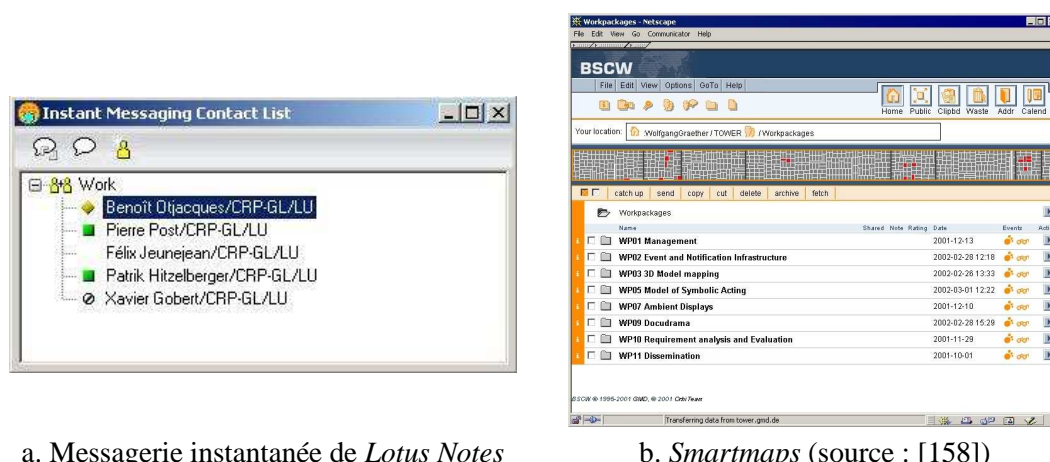
D'un point de vue général, il faut tout d'abord noter que contrairement à la visualisation des opérations réalisées sur des ressources partagées qui concernent principalement l'historique des données, la visualisation de la présence se focalise surtout sur la **représentation en temps réel du présent**. Par conséquent, les techniques proposées font le plus souvent référence à un contexte de communications synchrones (p.ex. 'chat') ou d'interactions en temps réel avec des ressources partagées (p.ex. 'shared whiteboard'). Cependant, le but habituel de la représentation de la présence diffère dans ces deux contextes d'application.

Dans le cas de communications synchrones, les visualisations de la présence d'autres personnes visent surtout à enrichir la communication textuelle pour la faire ressembler à une communication en face-à-face. Il s'agit alors de simuler les aspects non-verbaux de la communication, d'indiquer l'état de réceptivité ou de disponibilité ou encore de fournir des informations sur le rôle, la personnalité ou le statut des interlocuteurs. Par contre, dans le cas de la visualisation de la présence dans des environnements partagés, la raison d'être de la visualisation dépasse en général le cadre de la communication pure et simple. Il peut s'agir, par exemple, d'éviter le travail concurrent sur les mêmes objets ou, à l'opposé, de susciter des commentaires de la part de personnes actives sur les mêmes objets.

Une spécificité marquante des techniques de visualisation de la présence réside dans le fait qu'elles combinent de manière beaucoup plus intense que les autres deux types d'information : des informations purement factuelles et des informations plus subjectives, incitant à l'interprétation multiple. Cette caractéristique est particulièrement courante dans les systèmes de communication synchrone.

Les différentes approches diffèrent donc selon la nature de l'information visualisée. Nous distinguerons les visualisations selon le degré de subjectivité de l'information représentée. Ce degré de subjectivité est lié à l'ampleur des connotations implicites potentiellement associables à l'information.

Afin d'illustrer notre propos, quelques exemples de techniques de visualisation dans laquelle le concept de présence est important sont mentionnés ci-dessous. Elles sont classées en trois catégories associées à un degré de subjectivité faible, moyen ou fort.



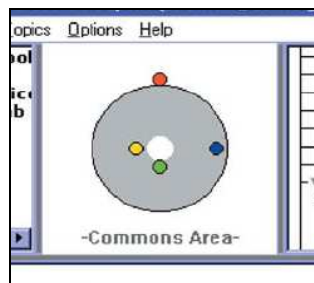
a. Messagerie instantanée de Lotus Notes

b. Smartmaps (source : [158])

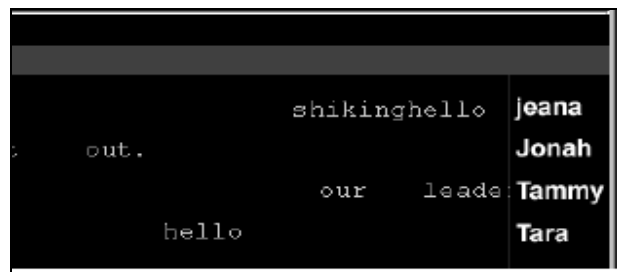
Fig. 49: Exemples de visualisation de présence avec composante subjective faible

Dans les visualisations de présence à composante subjective faible, seules des informations très factuelles sont représentées. Il s'agit essentiellement de données telles que le fait d'être connecté au système, le fait d'être disponible pour communiquer (cf. Fig. 49 a) ou encore le fait d'accéder à une ressource (cf. *Smartmaps*, Fig. 49 b, voir p. 106 pour une description détaillée). De plus, la technique de visualisation employée n'incite pas à de multiples interprétations subjectives.

Un premier niveau d'enrichissement de la représentation visuelle consiste à représenter des informations qui, bien que factuelles, incitent à l'interprétation subjective. Par exemple, dans la visualisation '*Babble*' (cf. Fig. 50 a), la représentation graphique de l'activité des personnes dans la conversation (cf. p. 97 pour une description détaillée) suscite des interprétations sur leur comportement social. De même, le système '*Fugue*' (cf. p. 97 pour une description détaillée) visualise le rythme de la conversation qui, lui aussi, peut être interprété de diverses manières. Cependant, comme nous l'avons déjà précisé, les informations qui sont représentées sont 'simplement' des informations collectées objectivement du système considéré, telles que le temps écoulé depuis la dernière action ou le rythme avec lequel les personnes tapent sur leur clavier.



a. *Babble* (source : [62])



b. *Fugue* (source : [171])

Fig. 50: Exemple de visualisation de présence avec composante subjective moyenne

De même, dans le système '*Tower world*' (Prinz et al., 2004), les objets partagés sont représentés dans un monde virtuel 3-D. Les participants sont représentés par des avatars qui sont positionnés sur les objets avec lesquels ils sont en train d'interagir. L'attitude de l'avatar indique le type d'opération en cours (p.ex. un avatar lisant un journal indique que l'utilisateur lit le document).

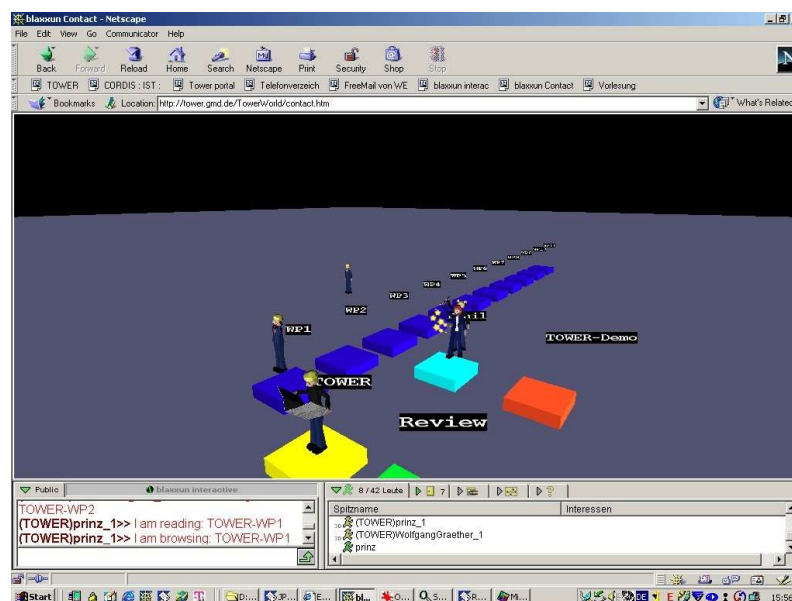


Fig. 51: Système '*Tower World*' (source: [158])

La troisième catégorie de visualisations de présence regroupe celles qui ajoutent à l'information factuelle d'autres données à composante subjective marquée. Ces dernières peuvent concerner, par exemple, l'humeur des intervenants ou l'ambiance dans le groupe. Elles peuvent être précisées :

- par les intervenants eux-mêmes dans une phase de configuration préalable à l'utilisation du système (p.ex. choix d'un avatar qui peut être représentatif de certains traits de la personnalité de l'utilisateur),
- par les intervenants pendant l'utilisation du système (p.ex. choix d'expressions dans le système 'ExMS'),
- ou encore par le système qui propose automatiquement des éléments de nature subjective (p.ex. expressions du personnage dans 'Comic Chat').



ExMS (source: [153])



Comic Chat (source: [108])

Fig. 52: Exemples de visualisation de présence avec composante subjective forte

Ces approches utilisent classiquement des représentations anthropomorphiques plus ou moins élaborées (p.ex. *emoticons*, *buddy icons*, *Comics*, *avatars*).

En conclusion, les techniques de visualisation de la présence dans un environnement partagé se caractérisent par :

- le fait qu'elles se réfèrent au présent,
- la représentation d'événements en temps réel,
- l'inclusion courante d'informations de nature subjective,
- les défis particuliers qu'elles suscitent en terme de connotations sociales.

4. Visualisations d'événements à caractère collectif

La dernière catégorie de visualisations que nous décrivons regroupe les techniques ayant pour but de représenter des informations associées à des événements présentant un intérêt collectif. Elle inclut notamment les visualisations d'agendas, de suivi de projet ou encore de flux de travail (*workflow*). Notons que l'intérêt collectif n'est pas synonyme de visibilité collective.

Ces visualisations se différencient des autres catégories par plusieurs éléments.

Tout d'abord, le facteur temporel joue un rôle majeur dans le type de données visualisées. En particulier, les représentations concernent non seulement des événements du passé et du présent mais aussi du futur, ce qui les différencie des autres catégories de visualisation qui portent le plus souvent sur le passé et / ou le présent.

Deuxièmement, les visualisations de ce type sont essentiellement destinées à supporter des tâches de planification ou de supervision. Elles ne sont donc pas directement concernées par des questions telles que l'identification des structures sociales (cf. visualisations de conversations) ou la représentation d'émotions (cf. visualisation de la présence).

Ensuite, la nature des relations entre événements successifs est spécifique. En effet, dans le cas des communications, les enchaînements temporels reposent habituellement sur le mécanisme ‘question – réponse’ ou ‘affirmation – commentaire’. Dans le cas des actions sur des ressources partagées, la temporalité est souvent associée à la problématique des versions successives d’un objet. Dans le cas de la visualisation de la présence, le caractère temporel se concentre sur la question du temps réel. Les événements à caractère collectif se relient l’un à l’autre par d’autres mécanismes tels que l’appartenance à un même contexte (p.ex. réunions d’un même projet) ou des contraintes de présence (p.ex. *workflow*). De plus, le parallélisme des événements est très courant (p.ex. tâches concurrentes en gestion de projet).

Nous présentons dans la suite quelques approches rencontrées dans la littérature pour visualiser des événements d’intérêt collectif. Cependant, comme les outils de gestion de projet ou de *workflow* constituent des thématiques en soi, leur étude approfondie dépasserait le cadre de cette revue de la littérature. Dès lors, nous nous limiterons au cas exemplatif de la visualisation de données associées à un agenda ou un calendrier. Notons que ces outils constituent une composante de base des logiciels de support au travail collaboratif.

Conformément à l’approche adoptée dans ce chapitre, nous structurerons notre état de l’art en nous basant sur les données qui sont à visualiser plutôt que sur les techniques. A ce titre, nous distinguerons deux groupes de travaux :

- ceux qui se limitent à visualiser les informations relatives à la description des événements et à leur positionnement temporel
- et ceux qui ajoutent d’autres informations à ces données de base (p.ex. préférences des personnes en terme de disponibilité).

Le premier ensemble de techniques regroupe donc les **représentations d’informations élémentaires sur la description et le positionnement temporel d’événements** tels que des réunions ou des rendez-vous. Nous considérerons ici à la fois des techniques de visualisation d’agendas personnels que d’autres destinées aux agendas de groupe.

La technique de visualisation d’agenda la plus courante utilisée dans ce cas repose sur l’utilisation de matrices dont les axes correspondent à deux niveaux de décomposition du temps (p.ex. mois – jour, jour – heure) et dont les cellules sont associées à un période de temps particulière (p.ex. jour, heure). Les événements sont associés à des glyphes positionnés dans la cellule correspondante.

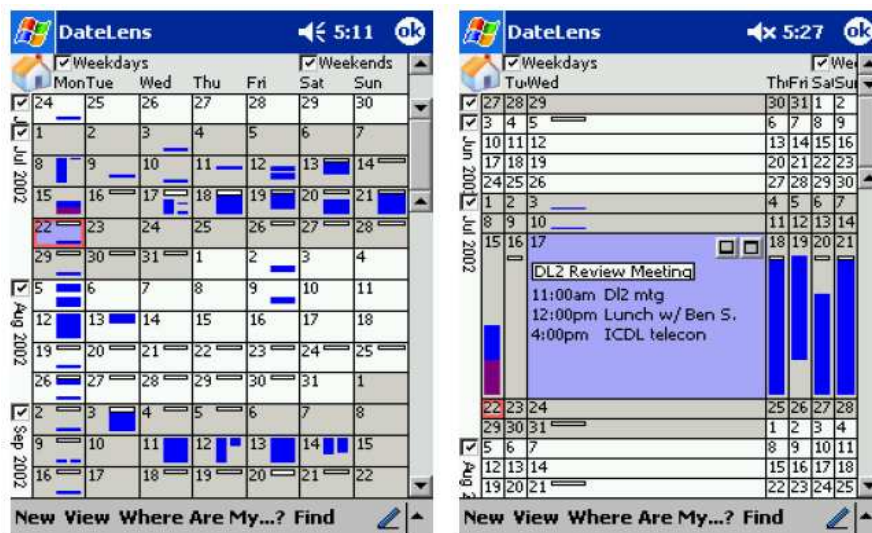


Fig. 53: Prototype ‘DateLens’ (source: [16])

Plusieurs auteurs ont proposé des améliorations à cette visualisation élémentaire. Par exemple, le prototype ‘DateLens’ (cf. Fig. 53) utilise le principe de la distorsion géométrique (œil de poisson – *fisheye view*) pour faciliter la lecture de certaines cellules (Bederson et al.,

2003 ; Bederson et al., 2004). Il propose une vue d'ensemble de plusieurs semaines (cf. Fig. 53, partie gauche) et une vue de détail focalisée sur un jour particulier (cf. Fig. 53, partie droite). Cette approche constitue une évolution récente adaptée aux terminaux mobiles (PDA) des travaux fondateurs de Furnas (1991).

La ligne du temps est une seconde technique classique de représentation de la temporalité d'événements. La visualisation '*Lifelines*' est un exemple souvent référencé en la matière (Plaisant et al., 1996). Elle consiste à représenter différents événements de la vie d'une personne sous la forme de lignes horizontales colorées pour donner une vue d'ensemble des activités de celle-ci au cours d'une période donnée (cf. Fig. 54). Bien que, à notre connaissance, elle n'ait pas été utilisée telle quelle pour visualiser un agenda proprement dit bien. Nous la mentionnons car elle est conceptuellement très proche de visualisations adoptées dans certains agendas de groupe (cf. p.ex. Agenda de groupe de *Lotus Notes*, Fig. 55).

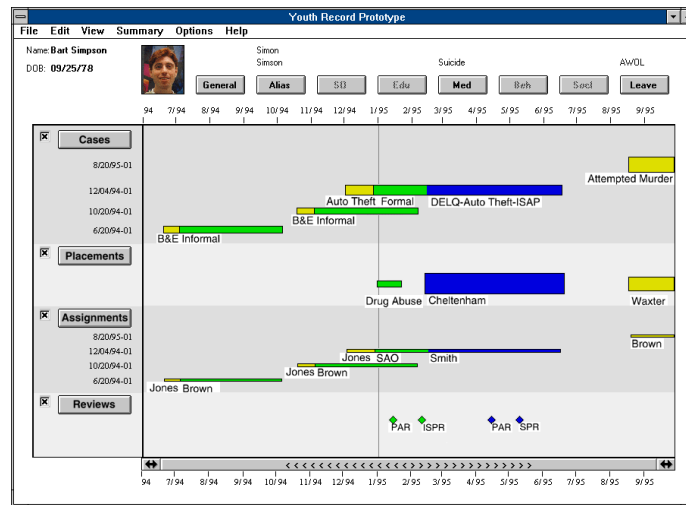


Fig. 54: Visualisation '*LifeLines*' (source: [156])

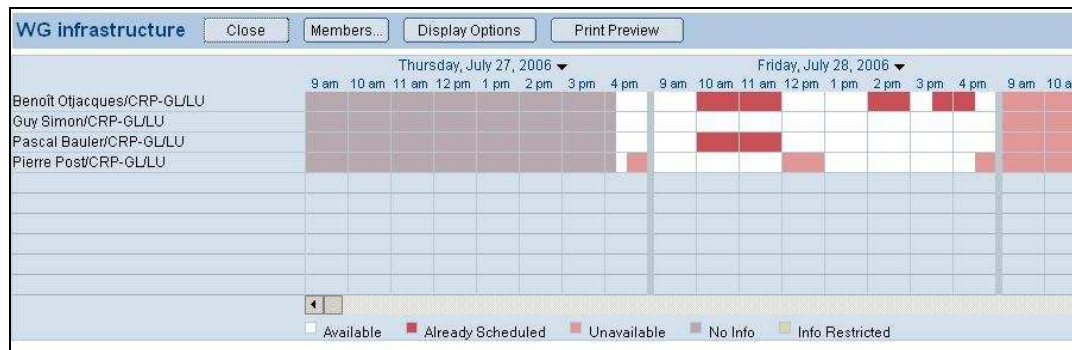


Fig. 55: Agenda de groupe de *Lotus Notes*

En relation directe avec les agendas, nous pouvons aussi mentionner le prototype '*TimeZoom*' (Dachselt et Weiland, 2006) qui propose une solution à la problématique bien connue de la visualisation du temps selon différentes granularités simultanément. Il se base sur des distortions multiples d'une ligne du temps pour permettre de se focaliser sur le détail de certaines périodes sans perdre de vue l'ensemble de la ligne du temps (cf. Fig. 56).



Fig. 56: Ligne du temps avec fonction '*TimeZoom*' (source: [42])

Une variante intéressante de représentation à base de ligne du temps a été proposée par le système ‘*Collaboration Information Gathering*’ (cf. Apperley et al., 2000 ; Masoodian et Budd, 2004). Il visualise un calendrier en tenant compte du décalage horaire entre différents lieux. L’axe vertical est exprimé en unités de décalage horaire. Un axe horizontal est associé à chaque endroit pour lequel des données sont à visualiser. Dans la Fig. 57, quatre axes sont représentés, associés aux villes d’Auckland, Sidney, Londres et Washington. Les événements (voyages en avion dans l’exemple) sont représentés par des lignes obliques s’ils traversent des fuseaux horaires et des lignes horizontales s’ils restent au même endroit.

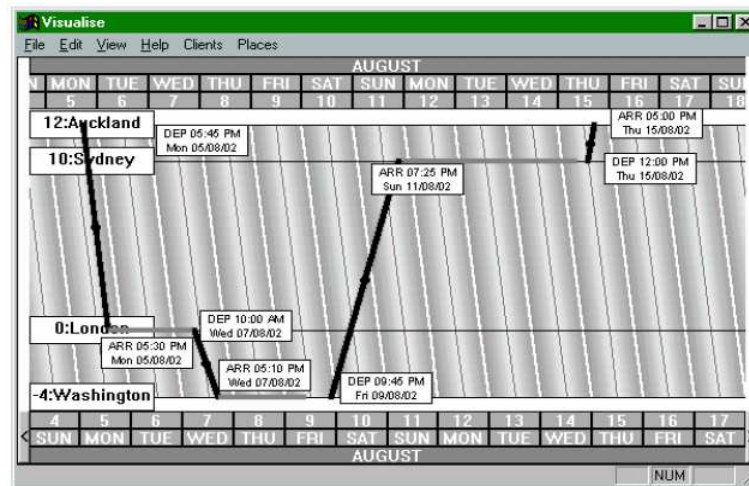


Fig. 57: Visualisation ‘CIG’ (source: [132])

Cette technique n’a pas été explicitement utilisée pour représenter des données de calendrier mais elle nous semblait utile à mentionner dans le cadre d’un état de l’art sur les moyens disponibles pour représenter le caractère temporel d’événements.

D’autres techniques originales de visualisation associées aux calendriers ont encore été proposées. Nous en mentionnerons deux, à titre d’exemples, dérivées du concept de spirale.

La visualisation ‘*Spiral Calendar*’ a été proposée en 1994 par une équipe du Centre de Recherche de Xerox à Palo Alto (Mackinlay et al., 1994). Elle positionne en spirale différentes vues semi-transparentes d’un calendrier. Chaque vue est un zoom sur une période particulière de la vue précédente placée en arrière plan (cf. Fig. 58).

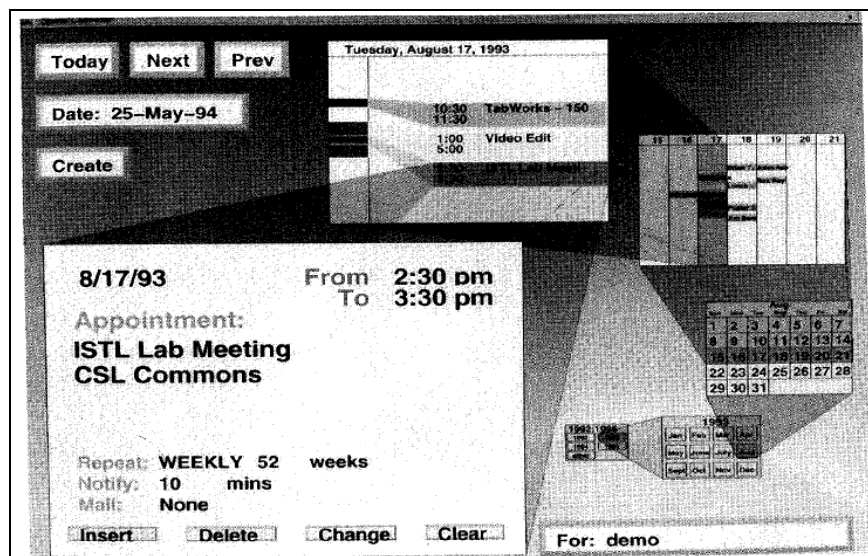


Fig. 58: Visualisation ‘Spiral Calendar’ (source: [127])

‘*SpiraClock*’ (Dragicevic et Huot, 2002) est une autre technique, plus récente, basée sur une visualisation en spirale. Dans ce cas, l’axe du temps s’enroule sous la forme d’une spirale (cf. Fig. 59). L’extérieur de la spirale est associée au temps présent. Plus on se rapproche du centre, plus on se déplace dans l’avenir. Les événements sont représentés par des secteurs colorés (p.ex. réunion aujourd’hui de 12:15 à 12:22 et réunion demain de 12:55 à 13:40).

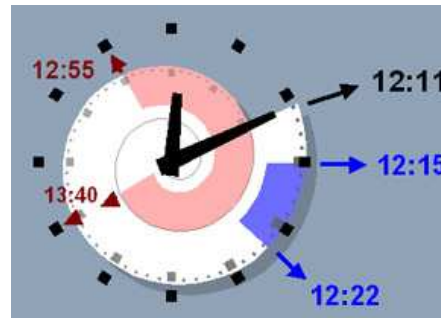


Fig. 59: Visualisation ‘*SpiraClock*’ (source: [192])

Si on se réfère aux données représentées, certaines visualisations **enrichissent les agendas avec des informations complémentaires** qui vont au-delà des données habituelles. Dans cette seconde catégorie de techniques, nous présenterons deux approches à titre exemplatif. La première ajoute des informations relatives aux préférences des personnes pour fixer des réunions. La seconde visualise la probabilité qu’une personne assiste à une réunion.

La technique ‘*Availability Bars*’ propose de visualiser, en plus des données classiques d’un agenda, les préférences des personnes pour planifier des événements (Faulring et Myers, 2006). Pour chaque plage horaire, on peut choisir sa préférence dans une gamme qui s’étend de ‘*exigé*’ à ‘*inacceptable*’. Les préférences sont représentées par un gradient de couleur (cf. Fig. 60a). En mode ‘agenda de groupe’ (cf. Fig. 60b), le système montre les préférences des membres du groupe et les disponibilités de ressources (p.ex. périodes de disponibilité des salles indiquées en bleu). Cette visualisation est destinée à favoriser la fixation des événements à des moments qui tiennent compte des préférences individuelles. Ajoutons que le système inclut un agent qui propose automatiquement des moments adéquats pour fixer une réunion.

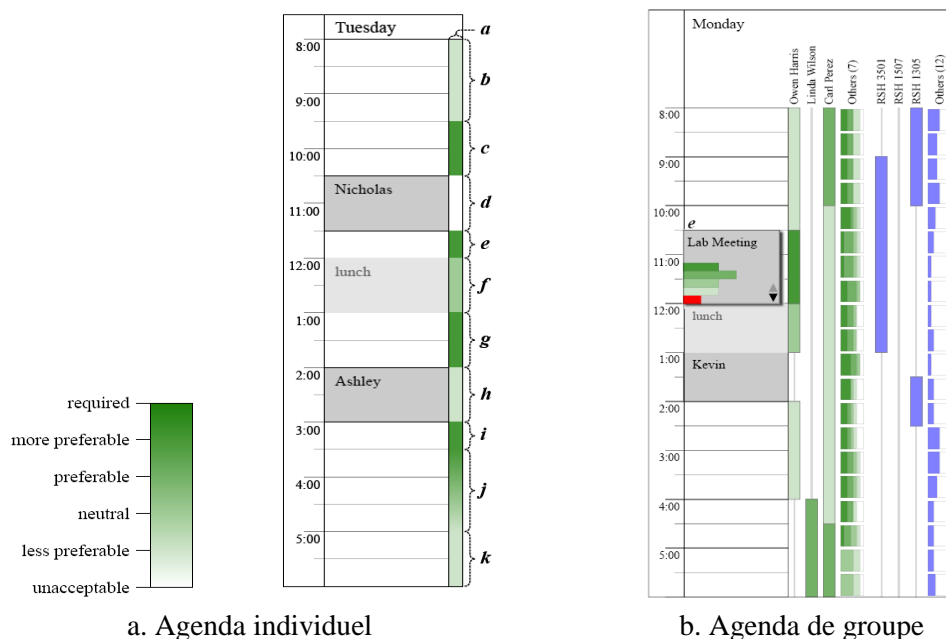


Fig. 60: Visualisation ‘*Availability Bars*’ (source: [64])

Pour leur part, Mynatt et Tullio (2001) proposent de visualiser plutôt la probabilité d'assister à une réunion. A partir d'un algorithme basé sur un réseau bayésien, ils calculent la probabilité qu'une personne assiste à une réunion planifiée dans son agenda. Cette probabilité est associée à un niveau de transparence (cf. Fig. 61). Les événements auxquels il est fortement probable que la personne assiste sont représentés par un rectangle quasi opaque (p.ex. DIEM) alors que ceux des quels elle sera probablement absente sont presque transparents (p.ex. ECL Seminar). Cette visualisation est accessible aux membres d'un département pour supporter la communication informelle en fournissant une meilleure perception de la disponibilité des collègues.

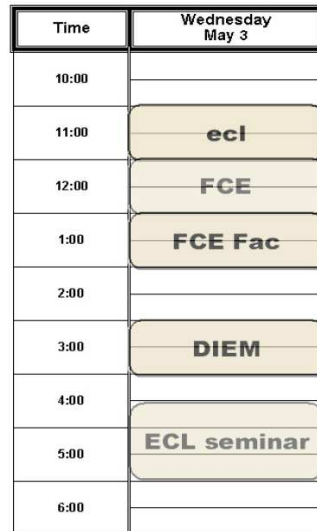


Fig. 61: Visualisation de la probabilité d'assister à un événement (source :[134])

Ces chercheurs proposent également de visualiser les facteurs principaux expliquant la probabilité de présence à un événement. Il s'agit en fait de représenter les variables principales du modèle bayésien. La taille des rectangles (p.ex. Priorité, Disponibilité) est proportionnelle à l'influence de la variable associée. Dans l'exemple de la Fig. 62, la probabilité que la personne concernée assiste à l'événement 'ecl' est de 80%, principalement parce qu'il n'y a pas de conflit de priorité et que la priorité de l'événement est haute. Le fait que la personne soit disponible intervient dans une moindre mesure.

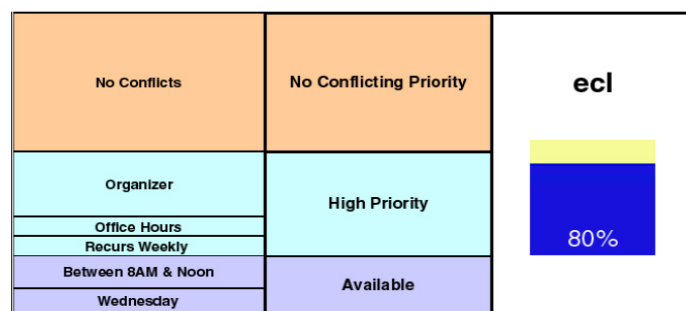


Fig. 62: Visualisation des facteurs explicatifs principaux de la probabilité de présence (source:[134])

En conclusion, les visualisations d'événements présentent des caractéristiques propres par rapport aux autres catégories, notamment :

- elles sont destinées à supporter des tâches de planification ou de supervision ;
- la visualisation concerne des événements du passé, du présent mais aussi du futur ;
- les relations entre les événements reposent sur des concepts spécifiques ;
- la représentation du temps à différentes échelles est particulièrement utile.

5. Visualisations combinées

Nous avons présenté un ensemble de techniques portant sur la visualisation d'objets associés à un groupe de personnes en situation de coopération. Afin de structurer notre démarche, nous les avons regroupées sur base du type de données qui sont représentées (conversations, événements ...). Cette approche est bien adaptée à l'étude de la littérature car la majorité des travaux recensés portent sur un seul type de données. Par exemple, les travaux sur la représentation de courriels ne s'intéressent habituellement pas aux actions sur des documents partagés ou sur les conversations par *chat* et réciproquement.

Cette approche différenciée est acceptable pour supporter les tâches relatives à un seul type de données, comme l'exploration de l'historique d'un document ou la recherche dans une liste de courriels envoyés à une date donnée. Cependant, elle présente des limitations importantes dans plusieurs situations.

Tout d'abord, dans la réalité, les échanges au sein d'un groupe reposent presque toujours sur plusieurs média de communication (p.ex. messagerie électronique, *chat*, téléphone, rencontres en face-à-face...). Seule une vue globale intégrée de l'ensemble de ces interactions permet donc de percevoir l'entière et la complexité des relations sociales entre les personnes. Si on se limite aux communications supportées par des technologies informatiques, cette vue d'ensemble, même partielle d'un point de vue social, constitue un pas en avant par rapport aux visualisations focalisées sur un seul média.

En second lieu, dans beaucoup de circonstances, une problématique, même limitée, est traitée en faisant appel à différents média de communication et différents outils informatiques. Par exemple, dans la plupart des cas, la visualisation de l'historique complet d'un document devrait inclure non seulement les versions stockées dans le système de gestion documentaire mais aussi, par exemple, les messages électroniques et les réunions qui le concernent. En fait, très peu de problématiques font appel à un seul moyen de collaboration.

Enfin, dans certains cas, l'utilisateur a besoin de percevoir toute la complexité de la situation courante. Cette problématique se pose notamment pour les managers qui doivent gérer un porte-feuille d'activités et qui doivent en permanence arbitrer entre les priorités de leurs tâches. Ici encore, une vue limitée à un seul type de données oblige l'utilisateur à naviguer entre plusieurs représentations partielles de la situation, ce qui conduit à une surcharge cognitive et à un risque réel d'oublier certaines informations ou de ne pas percevoir l'interdépendance entre des événements.

Une seconde limitation observée dans la plupart des prototypes décrits dans la littérature concerne le fait que la majorité d'entre eux sont incapables de supporter plusieurs technologies. A titre d'exemple, les systèmes expérimentaux décrits dans la littérature ne permettent presque jamais de visualiser de manière intégrée des courriels échangés via différentes technologies (p.ex. *Microsoft Outlook*, *Lotus Notes*, systèmes de *WebMail*) bien que la plupart des personnes disposent de nos jours de comptes dans plusieurs systèmes de messagerie. De même, à notre connaissance, peu de systèmes permettent de visualiser ensemble les opérations effectuées sur un document dans un système de gestion documentaire et celles effectuées sur le même document stocké sur le disque dur local de son ordinateur.

La conséquence finale de ces limitations est que la plupart des propositions n'offrent qu'une vue très partielle de l'environnement dans lequel se trouve l'utilisateur. La double limitation : visualisation différenciée des types de données et des technologies sous-jacentes pose problème si on veut présenter à l'utilisateur une vue plus globale de l'environnement électronique dans lequel il se meut.

Cette problématique a été identifiée par certains auteurs. Par exemple, l'approche 'Soylent' (Fisher et Dourish, 2004) se base uniquement sur les archives de courriels. Les auteurs indiquent que le prototype est conçu de manière à pouvoir intégrer d'autres sources d'information mais ce n'est pas réalisé en pratique. De plus, dans leur approche égo-centrique, ils construisent une base de données locale dans laquelle la technique de visualisation va puiser. L'accès en temps réel aux données issues de sources multiples n'est pas envisagé.

Quelques auteurs ont néanmoins proposé des réponses à ces limitations en essayant de combiner les informations de différents types. Nous présentons ci-après certains de ces travaux.

Lorsque différents types de données doivent être visualisés dans un environnement informatique partagé, l'approche classique consiste à juxtaposer des visualisations associées à chacun d'entre eux. Dans plusieurs plates-formes électroniques de coopération, une vue sur les documents partagés est disponible, de même qu'une autre sur les conversations par messagerie instantanée ou sur les messages électroniques mais le système ne propose habituellement pas une vue d'ensemble combinant toutes ces informations. Par exemple, Kubicki et al. (2006) proposent une vue globale d'un projet de construction qui juxtapose un diagramme de Gantt, un graphe et une structure de liste (cf. Fig. 63).

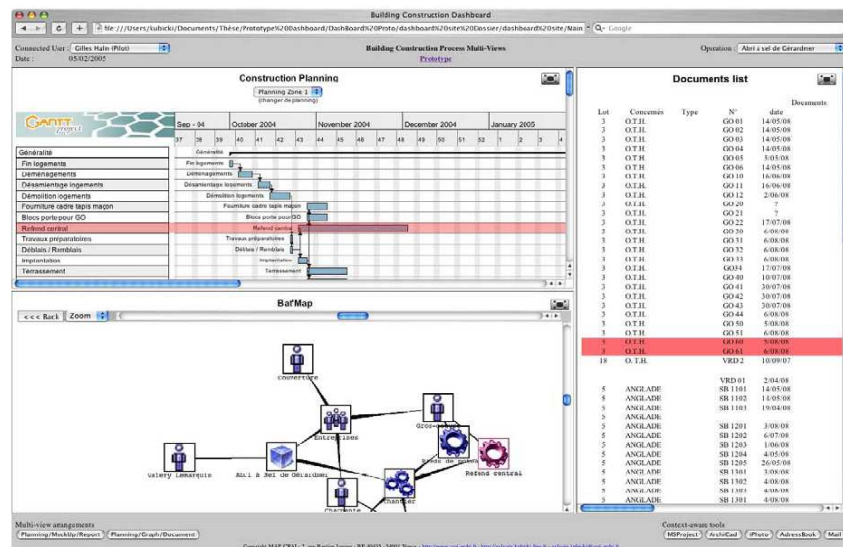


Fig. 63: Juxtaposition de vues multiples d'un projet de construction (source: [107])

Une seconde approche également basée sur le principe de juxtaposition consiste à combiner un outil de communication synchrone avec un outil d'interaction avec un document. Cet ensemble de fonctionnalités est principalement destiné à supporter des discussions en temps réel sur des documents partagés.

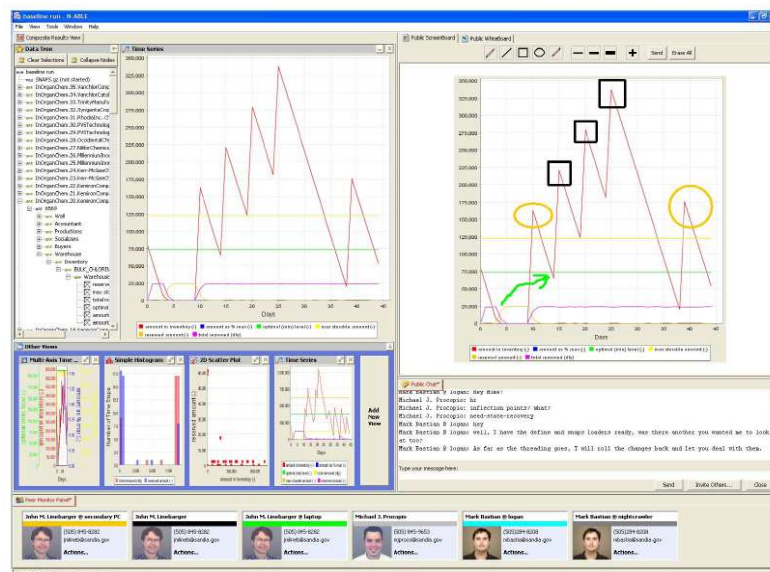


Fig. 64: Système 'N-ABLE' (source: [118])

Le système '*N-ABLE*' (Linebarger et al., 2006) illustre bien cette approche (cf. Fig. 64). Il combine principalement un indicateur de présence des personnes via des photos, un outil de messagerie instantanée, un outil de partage d'écrans avec possibilité d'annotations, un écran blanc partagé et un système de transfert de fichiers.

La juxtaposition d'informations hétérogènes dans un même écran est une première solution pour donner une vue d'ensemble d'une situation. Cependant, elle ne peut être considérée comme une vue intégrée des données basée sur une technique de visualisation homogène.

La véritable intégration des données hétérogènes dans une représentation unique est peu présente dans la littérature.

Le système VCV (*Virtual Communications Visualization*) proposé par Beale et al. (2001) en est un des rares exemples que nous connaissons. Ces auteurs cherchaient à aider des collaborateurs et des superviseurs à avoir un aperçu de l'efficacité des communications au sein d'un groupe. Une des idées sous-jacentes à cette approche est qu'en montrant aux participants d'une communication comment elle se développe, on espère qu'ils pourront tirer plus d'utilité de l'information et qu'ils pourront comprendre plus facilement le contexte de la communication.

Une des originalités de leur approche est que le système VCV est capable de représenter sous une forme complètement intégrée des conversations menées par le biais de différents médias électroniques : *chat*, vidéoconférence, messagerie électronique... (cf. Fig. 65) Dans l'exemple de la Fig. 65, une vidéoconférence est suivie de l'édition collaborative d'un bloc notes. Dans la partie supérieure de la représentation graphique, une ligne du temps globale (*Global Timeline*) est affichée. Elle couvre une période de plusieurs mois. Une seconde ligne du temps (*Secondary Timeline*) est également présentée pour permettre de visualiser les détails associés à une période plus courte (quelques heures). La fenêtre de conversation (*Conversation Window*) visualise la conversation proprement dite avec ses intervenants. Un élément est particulièrement digne d'être remarqué dans la représentation VCV. Des symboles graphiques similaires sont associés à des événements appartenant à des niveaux conceptuellement différents. On peut en effet distinguer d'une part l'utilisation de médias de communication (*chat*, vidéoconférence) et d'autre part des opérations associées à des documents dans le contexte de l'utilisation du *notebook* (*save*, *lock*).

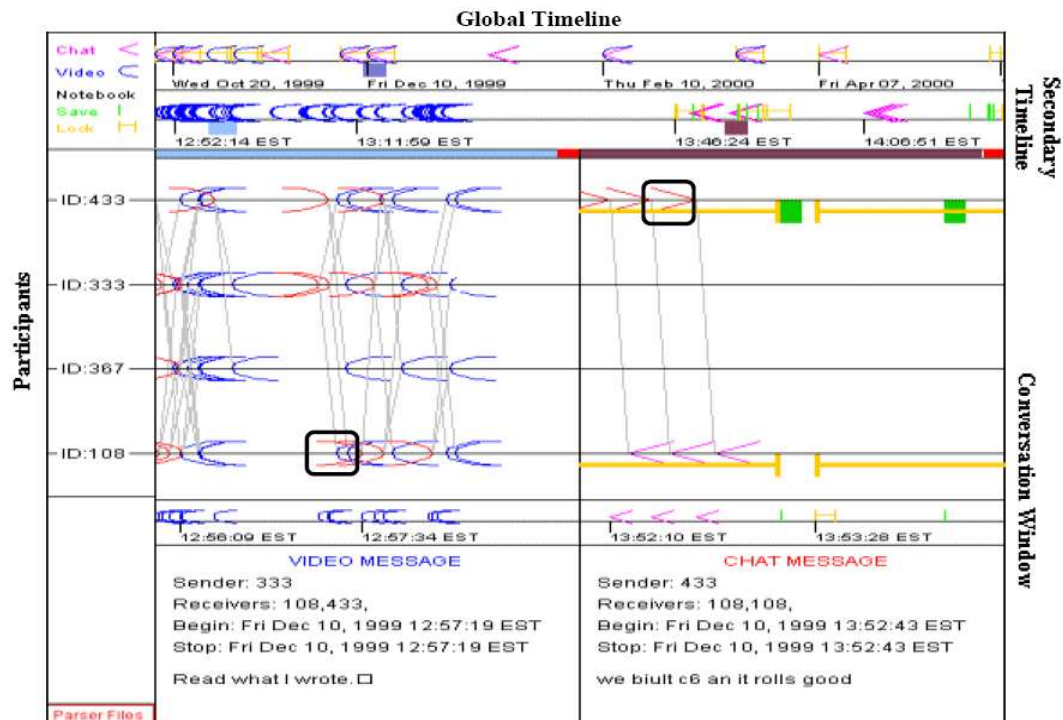


Fig. 65: Représentation intégrée VCV (source : [15])

Pour permettre une visualisation intégrée, le système VCV repose sur la modélisation des communications comme un ensemble de quatre éléments de base: un expéditeur, un ou plusieurs destinataire(s), le temps (celui de l'expédition et éventuellement celui/ceux de la réception) et le contenu du message. Le système VCV est capable d'importer des données comportant ces attributs minimaux.

Begole et al. (2002) ont également combiné des informations venant de plusieurs sources mais en se concentrant sur une troisième problématique : l'analyse des rythmes d'activité dans des groupes distribués. Ils ont collecté des données provenant de l'interaction entre l'utilisateur et son ordinateur (activité au clavier et avec la souris), d'un agenda en ligne et de la messagerie électronique.

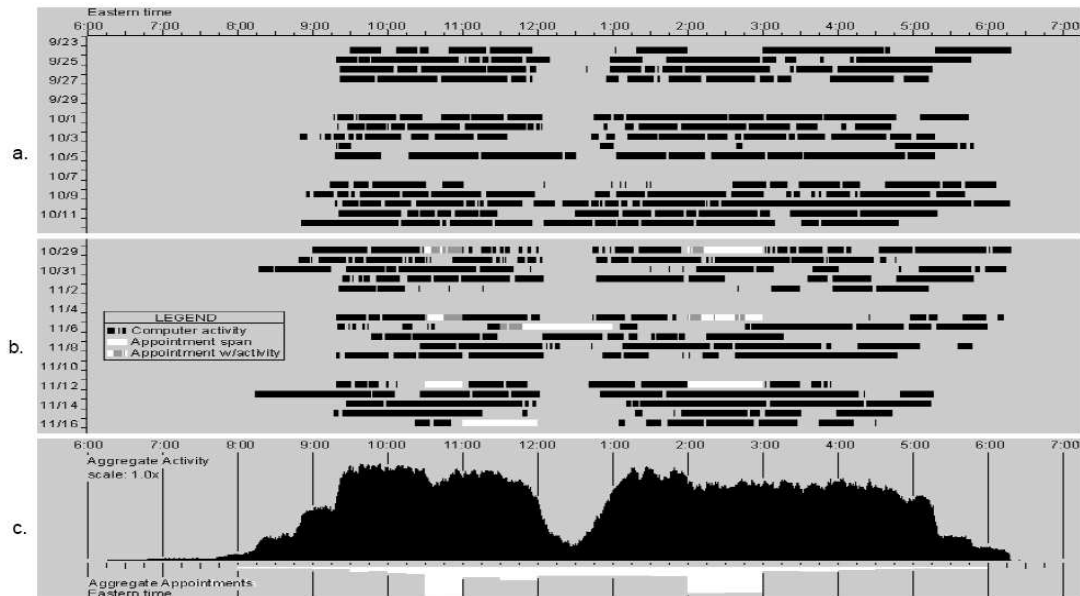


Fig. 66: Visualisation de l'activité (source: [18])

La Fig. 66 illustre la représentation graphique de ces données¹⁹. Le temps est associé à l'axe horizontal en ce qui concerne les heures et à l'axe vertical pour les jours. La partie supérieure de la visualisation (cf. Fig. 66a) montre l'activité de l'utilisateur au cours de trois semaines. La partie centrale (cf. Fig. 66b) combine la représentation de l'interaction avec l'ordinateur et celle des réunions. Enfin, la partie inférieure (cf. Fig. 66c) illustre la distribution de l'ensemble des activités au cours d'une journée.

Ces deux derniers exemples sont assez classiques en termes de techniques de visualisation. Leur intérêt majeur vient du caractère hétérogène des données qu'ils représentent. Dans ce cas, une problématique majeure réside dans la collecte des données brutes et dans leur transformation dans un modèle homogène cohérent qui pourra servir de base à la représentation graphique.

A notre connaissance, peu de logiciels proposent une représentation intégrée d'informations du même type reposant sur des technologies différentes. Nous citerons néanmoins le client de messagerie instantanée libre source '*GAIM*' (2006). Il supporte différents protocoles et permet de discuter via la même interface graphique avec des personnes utilisant différentes technologies : *AOL Instant Messaging*, *ICQ*, *MSN Messenger*, *Yahoo !*, *IRC*, *Lotus Sametime*...

¹⁹ Notons que la visualisation combinée d'informations relatives aux activités d'un individu pose des questions en termes éthiques et juridiques (cf. protection de la vie privée).

6. Conclusions

Nous avons montré dans ce chapitre que la visualisation de données associées à un contexte collaboratif a déjà fait l'objet d'un certain nombre de travaux scientifiques bien qu'elle puisse apparaître comme une thématique de recherche très ciblée. Nous avons décrit quelques travaux que nous jugeons être représentatifs de cette problématique de recherche. Plusieurs enseignements peuvent être tirés de cette étude.

Tout d'abord, nous attirons l'attention sur le fait qu'aucune des techniques décrites dans ce chapitre et, à notre connaissance aucune des techniques existantes, n'a pour ambition de visualiser toutes les composantes d'une situation coopérative. Aucune ne permet de visualiser toutes les données qui y sont relatives, indépendamment des outils techniques associés ou des fonctions²⁰ qu'ils assurent dans la coopération.

Nous avons également observé que peu d'auteurs se sont confrontés à la question de représenter des données hétérogènes issues de diverses sources supportées par des technologies multiples. Le manque d'homogénéité dans les technologies et les formats de données rendent difficiles la conception et l'implémentation d'approches intégrées. Néanmoins, en pratique, cette question se pose de manière cruciale. En effet, la plupart des situations coopératives mettent en œuvre des outils différents ayant des buts différents.

Dans notre présentation, nous avons choisi de différencier les techniques selon **la nature des données visualisées**. Notons que dans notre acceptation, ce concept fait référence à la sémantique des données plutôt qu'à leur structure. Ils ont donc été regroupés en quatre catégories :

- ⇒ les visualisations de communications (électroniques) ;
- ⇒ les visualisations d'actions sur des ressources partagées ;
- ⇒ les visualisations de la présence d'acteurs ;
- ⇒ les visualisations d'événements d'intérêt collectif.

Cependant, l'analyse de ces différentes techniques nous conduit également à identifier d'autres critères importants pouvant servir de base à une classification.

Nous commencerons par nous intéresser au **but de la visualisation**.

Certains systèmes ont pour vocation de supporter une certaine composante de la coopération en vue de la rendre plus efficace, plus conviviale ou mieux traçable. Dans ce contexte, la visualisation sert de support pour faciliter ou améliorer la réalisation d'une tâche ou d'un groupe de tâches. A titre d'exemple, citons les systèmes '*Chat Circle*', '*Comic Chat*' ou '*Fugue*' qui ont pour objectif d'améliorer un système de *chat* en y ajoutant une représentation graphique d'éléments habituellement absents de tels systèmes. Plus précisément, la visualisation est alors employée pour créer ou augmenter le niveau de conscience de groupe (*awareness*) dont l'importance dans une situation coopérative a été démontrée par de multiples auteurs (cf. p. 21).

Dans d'autres cas, la fonctionnalité de visualisation est conçue pour permettre une analyse de la coopération. Par exemple, les systèmes '*Clique Panel*', '*Social Network Fragment*', '*Authorlines*' ou '*CodeSaw*' appartiennent à cette catégorie. Ces approches illustrent également le fait que la fonction d'analyse requiert très souvent la représentation d'un historique des données.

La relation entre le système de visualisation et l'application informatique qui supporte la coopération est un autre critère qui peut servir à différencier les approches rencontrées dans la littérature. Ce critère peut s'exprimer en termes de **niveau d'intégration** des deux systèmes. Deux grandes options sont possibles : l'intégration ou l'indépendance des applications.

²⁰ Communication synchrone, partage de documents, historique des visites dans un espace partagé...

Dans le premier cas, la fonctionnalité de visualisation fait partie intégrante du système de support à la coopération. Les systèmes ‘*ArchiChronos*’, ‘*Babble*’ ou ‘*Palace*’ illustrent cette approche. L’interface utilisateur combine des fonctionnalités de visualisation et de support à la coopération. Par exemple, ‘*Babble*’ intègre la visualisation des participants au *chat* dans l’interface permettant d’écrire les messages.

Dans le second cas, le module de visualisation est distinct des modules coopératifs. Bien qu’il accède à des données issues de fichiers ou de bases de données associées au système coopératif, il se présente à l’utilisateur comme une application distincte. La représentation de fichiers d’archives (‘*log files*’) en est un exemple classique. Parmi les systèmes référencés dans notre état de l’art, nous pouvons par exemple mentionner les applications ‘*History Flow*’ et ‘*Newsgroup Crowds*’.

A ce propos, une petite digression nous semble nécessaire. Certains systèmes se prétendent intégrés alors qu’ils ne font que juxtaposer des informations et des fonctionnalités disjointes. C’est par exemple le cas de certains portails accessibles via web. Ils se contentent de diviser l’espace d’affichage en zone indépendantes et d’attribuer ces zones à des modules applicatifs distincts. Pour notre part, nous considérons que l’intégration exige la mise à disposition de l’utilisateur d’une interface unique dans laquelle la production, la mise à jour et la visualisation des données sont synchronisées en temps réel de manière permanente.

L’utilisateur cible peut également être un élément pouvant être utilisé en vue de distinguer les techniques de visualisation. Certaines approches sont principalement destinées à un utilisateur prenant part à l’activité coopérative. L’utilisateur pourra donc notamment voir sa propre activité. Il connaît également l’ensemble ou une partie des autres participants, ainsi que le contexte dans lequel la coopération a lieu. Les systèmes ‘*Chat Circle*’ ou ‘*Date Lens*’ appartiennent à cette catégorie.

D’autres techniques sont conçues pour être utilisées essentiellement par des observateurs externes à la situation coopérative. On peut citer comme exemples les systèmes de supervision du développement de logiciel ‘*Code Connections*’ ou ‘*Code Saw*’ ou le système ‘*History Flow*’ de visualisation de l’historique des opérations sur une page web. Dans cette approche, les questions relatives à la protection des données à caractère personnel se posent de manière plus aiguë. Ces systèmes peuvent notamment être utilisés à des fins de compréhension d’une situation particulière, de surveillance de l’interaction entre les acteurs, de détection de comportements dangereux ou abusifs ou de mise en évidence de processus implicites.

La revue de la littérature relative aux techniques de visualisation d’activités coopératives présentée dans ce chapitre ajoute une seconde pièce à notre puzzle en vue de proposer une taxonomie de ces techniques.

Nous avons commencé par décrire dans un chapitre antérieur (cf. p. 39), un ensemble de taxonomies de techniques de visualisation sans nous focaliser sur l’aspect coopératif des données représentées. Ceci nous a permis d’identifier les critères retenus par d’autres chercheurs et d’identifier des propriétés génériques des taxonomies.

Ensuite, dans ce chapitre, nous nous sommes focalisés sur les systèmes coopératifs. Nous avons abordé le problème sous un autre angle. A lieu de nous baser sur des réflexions conceptuelles relatives à l’élaboration d’une taxonomie, nous avons plutôt recensé des travaux décrivant des systèmes concrets et nous avons tenté d’identifier ce qui les distingue ou les rassemble.

Cette combinaison d’approches descendante (*top-down*) et ascendante (*bottom-up*) fut particulièrement fructueuse. En effet, elle oblige à associer une réflexion théorique sur le concept même de taxonomie et une étude pragmatique des systèmes existant sur le terrain.

Nous sommes maintenant à même de proposer notre taxonomie des techniques de visualisation de données à caractère coopératif. Cela fera l’objet du prochain chapitre.

Chapitre 5 : Taxonomie des techniques de visualisations de la coopération

« Pour le savant, croire la science achevée est toujours une illusion aussi complète que le serait pour l'historien de croire l'histoire terminée. »

L. V. de Broglie

1. Cadre de travail

Dans un chapitre précédent, nous avons analysé différentes taxonomies proposées dans le domaine de la visualisation d'information (cf. p. 39). Ensuite, nous avons étudié divers systèmes destinés à visualiser des données relatives à des activités coopératives (cf. p. 87). Ces travaux nous ont permis de mettre en évidence qu'il ne semble pas exister de taxonomies spécifiquement focalisées sur la visualisation d'informations associées à un contexte coopératif.

Cette lacune est particulièrement dommageable si on se réfère à l'utilité de disposer de taxonomies dans toute discipline scientifique, ainsi que le précise H. Simon (1969) : « *the first step to understanding any set of phenomena is to develop a taxonomy* ».

Dans toute matière scientifique, les taxonomies sont nécessaires pour un ensemble de raisons, parmi lesquelles les plus importantes sont probablement :

- permettre une revue systématique et rigoureuse des techniques et idées actuelles,
- positionner des travaux spécifiques dans le champ de recherche tout entier,
- clarifier les concepts du domaine, ce qui est un étape nécessaire avant d'élaborer une nouvelle théorie,
- identifier les similarités et les différences entre les résultats de recherche,
- faire ressortir des nouvelles directions de recherches,
- mettre en évidence les imperfections dans les propositions de recherche.

Ce chapitre se positionne donc comme une proposition pour combler partiellement cette lacune dans le domaine des techniques de visualisation d'activités coopératives.

2. Approche générale

2.1 Définitions

La taxonomie proposée dans ce chapitre a été dénommée : « *Interactions Visualization Framework* » (IVF).

Le modèle IVF se focalise tout particulièrement sur les « actions » se produisant dans un contexte de coopération électronique. Son but n'est donc pas de modéliser tous les aspects d'une situation de coopération réelle.

D'un point de vue conceptuel, nous avons choisi de modéliser une activité coopérative comme un ensemble d'*interactions* entre des *entités* (cf. p. 29). Ces entités peuvent se présenter sous diverses formes : agissantes (p.ex. personnes) ou passives (p.ex. documents) ; individuelles (p.ex. une seule personne) ou collectives (p.ex. une équipe) ... De même, de multiples types d'interactions peuvent être rencontrés : communications inter-personnelles, échanges de documents, créations ou modifications de ressources partagées...

Nous avons également précisé que nous nous limitons à l'étude des interactions supportées par des outils informatiques.

Il peut être utile à ce stade de rappeler les définitions utilisées dans notre modèle.

- ⇒ Une *entité* est une personne ou une ressource prenant part à une situation coopérative.
- ⇒ Une *interaction* se produit entre des entités et constitue un échange d'information entre elles.
- ⇒ Une *interaction électronique* se produit entre des entités et constitue un échange d'information reposant sur l'utilisation d'outils informatiques.

Après avoir défini les composantes de la coopération considérées dans le modèle, il est également indispensable de préciser ce que nous entendons par 'technique de visualisation'. En effet, le modèle IVF a pour objet de classifier des techniques de visualisation plutôt que des situations de coopération (même si les deux volets sont évidemment liés).

**Une technique de visualisation est une unité élémentaire
de représentation de données sous forme graphique.**

Les termes de cette définition méritent d'être explicités.

Tout d'abord, une technique de visualisation est un mode de représentation de données. Dans le cas présent, les données consisteront en une collection d'entités et d'interactions.

En second lieu, la représentation doit se présenter sous forme graphique, c.-à-d. perceptible par la vue. La représentation doit donc être conçue pour être interprétée par des êtres humains, ce qui exclut, par exemple, celles destinées aux ordinateurs.

Enfin, la représentation doit être élémentaire, dans le sens où elle doit apparaître comme un objet conceptuellement unitaire. Le terme 'unitaire' fait référence à la relation existant entre les données et les variables visuelles utilisées dans la représentation. Par exemple, un arbre et un *treemap* constituent deux relations différentes entre données et variables visuelles. De notre point de vue, il s'agit donc de deux techniques de visualisation distinctes, même si elles sont présentées dans la même fenêtre d'un logiciel.

Nous appellerons 'composition de techniques de visualisation' une association de plusieurs techniques de visualisation. L'association peut être spatiale, c.-à-d. qu'elle juxtapose plusieurs techniques de visualisation dans un même espace d'affichage (ex. *technique 'linking and brushing'*). L'association peut aussi être temporelle, c.-à-d. qu'elle permet de passer successivement d'une technique à une autre. Enfin l'association peut combiner ces deux aspects et être spatio-temporelle.

Cette remarque se justifie par le fait que, dans la pratique, la plupart des logiciels de visualisation combinent plusieurs techniques de visualisation. Dans l'approche IVF, un logiciel donné ne constitue donc pas un objet d'analyse pertinent en soi.

2.2 Méthodologie de conception de la taxonomie IVF

D'un point de vue méthodologique, le modèle IVF s'inscrit comme la suite logique des réflexions sur les taxonomies présentées précédemment. En particulier, notre méta-taxonomie des techniques de visualisation d'information a servi de base à sa conception.

Le modèle fut développé de manière itérative. Le cycle de réflexion suivant fut instancié à de multiples reprises.

1. Réflexion théorique sur les propriétés d'une technique de visualisation
2. Analyse des taxonomies existantes
3. Proposition de propriétés et de leurs valeurs autorisées
4. Confrontation des propriétés avec des techniques de visualisation réelles
5. Modification des propriétés et des valeurs autorisées

Certaines propriétés furent reprises telles quelles de taxonomies préexistantes. D'autres sont issues de travaux antérieurs mais furent adaptées. D'autres enfin, résultent de nos réflexions sur les objets à caractériser.

2.3 Principes directeurs de la taxonomie IVF

Le modèle IVF a pour vocation de classer les techniques de visualisation utilisées pour représenter des situations dans lesquelles des entités sont en interaction.

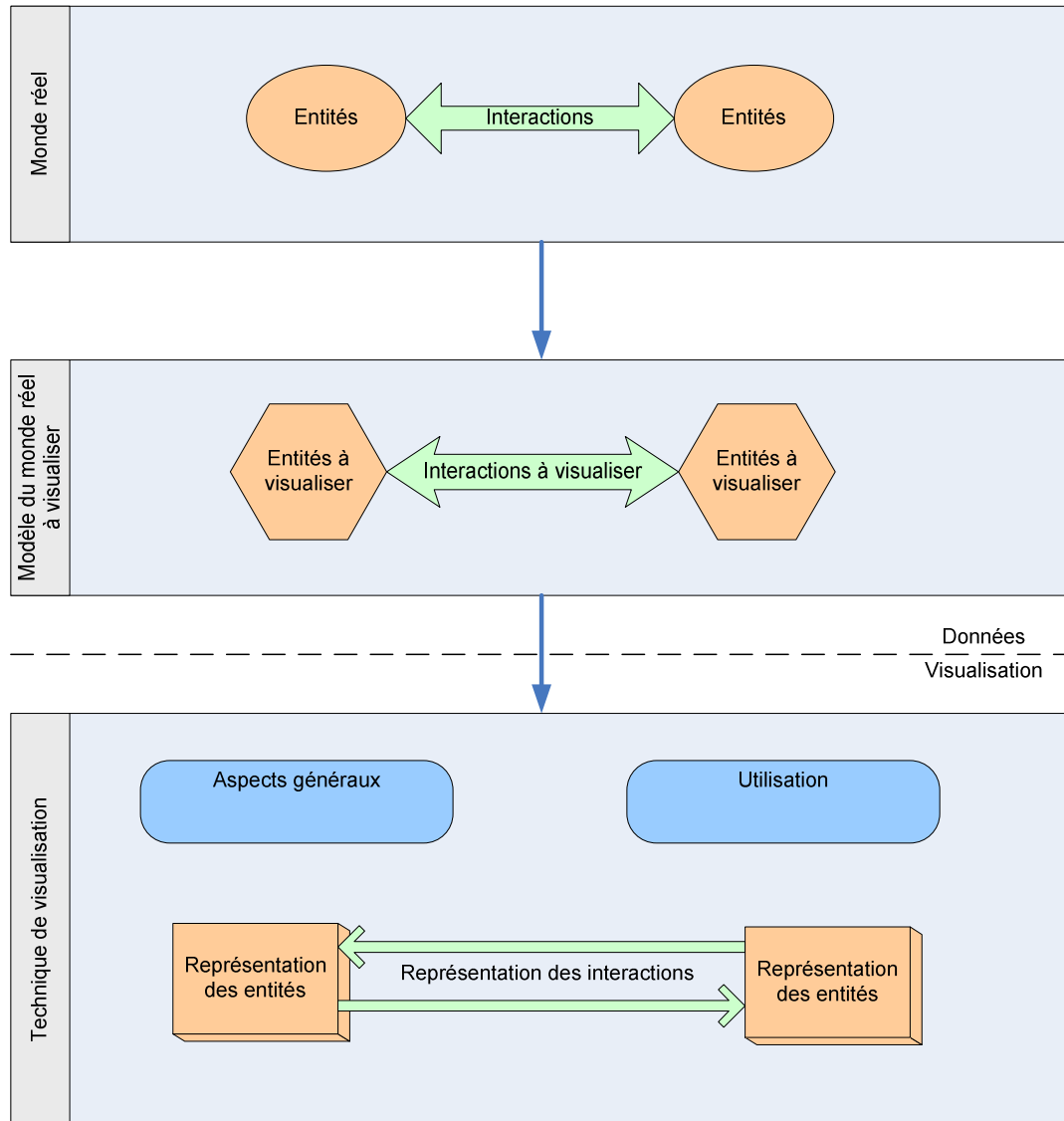


Fig. 67: Principes directeurs du modèle IVF

La première différenciation à mettre en évidence concerne la distinction entre les données à visualiser et la technique utilisée pour cette visualisation. En effet, afin de pouvoir caractériser une technique de visualisation, il faut d'abord être capable de spécifier la nature des données à représenter.

Si on s'intéresse de plus près à ces données, on réalise qu'elles ne constituent qu'un modèle de la situation observée dans le monde réel. Il faut donc, logiquement, commencer par **décrire la situation réelle** au moyen d'un premier ensemble de propriétés. A titre d'exemple, on peut mentionner le besoin de spécifier si la situation de coopération concerne des entités distantes ou localisées au même endroit.

Ensuite, un second ensemble sera nécessaire pour **caractériser le modèle du monde réel** qui servira de base à la visualisation. Par exemple, une propriété pourra être dédiée à spécifier si la seule présence des entités est représentée ou si certains de leurs attributs le sont également.

En ce qui concerne la **technique de visualisation**, différentes composantes doivent être décrites par un ensemble de propriétés spécifiques. En premier lieu, il faut décrire la technique de visualisation en général. Par exemple, s'agit-il d'une visualisation à une, deux ou trois dimensions ? Ensuite, il est nécessaire de préciser le cadre d'utilisation de la visualisation. Par exemple, la visualisation est-elle destinée à l'analyse d'une situation de coopération ou à améliorer la qualité des interactions ? Enfin, un dernier ensemble de propriétés aura pour but de spécifier la manière dont les composantes de base de notre modèle – les *entités* et les *interactions* – sont représentées graphiquement. Par exemple, la variable rétinienne '*couleur*' est-elle utilisée dans la représentation des interactions ?

En résumé, les propriétés associées aux deux niveaux '*Monde réel*' et '*Modèle du monde réel à visualiser*' permettent de répondre à la question : '*Quels objets doivent être visualisés ?*' Les propriétés associées au niveau '*Technique de visualisation*' s'attachent à répondre aux questions : '*Comment les visualiser ?*' et '*Dans quel contexte sont-ils visualisés ?*'

Dans la suite de ce chapitre, les propriétés relatives à ces différents aspects seront décrites de manière théorique et illustrées par divers exemples concrets.

2.4 Notations

Afin de faciliter la compréhension du modèle de manière non ambiguë, nous avons adopté un formalisme simple mais rigoureux pour décrire les différentes propriétés qui le composent.

Dans la suite du texte, les propriétés seront référencées par des caractères en italique et en gras (cf. **Propriété A**) alors que les valeurs des propriétés utiliseront des caractères italiques (cf. *valeur₁*).

Certaines propriétés IVF sont univaluées alors que d'autres sont multi-valuées.

Les propriétés uni-valuées adopteront la notation suivante :

$$\textbf{Propriété A} : \{ \textit{valeur}_1 - \textit{valeur}_2 - \textit{valeur}_3 \}$$

Cette notation doit être interprétée comme suit. Pour une technique de visualisation donnée, la **propriété A** peut se voir affecter une seule des valeurs : '*valeur₁*', '*valeur₂*', '*valeur₃*'.

Cette situation correspond à un critère 'à valeur unique' (cf. p. 85)

Les propriétés multi-valuées adopteront la notation suivante :

$$\textbf{Propriété A} : \{ \textit{valeur}_1, \textit{valeur}_2, \textit{valeur}_3 \}$$

Cette notation doit être interprétée comme suit. Pour une technique de visualisation donnée, la **propriété A** peut se voir affecter toute combinaison des valeurs : '*valeur₁*', '*valeur₂*', '*valeur₃*'.

Des expressions logiques simples peuvent être construites avec ce formalisme. Par exemple, si la **Propriété A** peut prendre ou bien la *valeur₁* ou une combinaison de valeurs de l'ensemble {*valeur₂*, *valeur₃*}, on exprimera cette contrainte comme suit :

$$\textbf{Propriété A} : \{ \textit{valeur}_1 - \{ \textit{valeur}_2, \textit{valeur}_3 \} \}$$

3. La taxonomie IVF

3.1 Propriétés descriptives de la situation réelle

Le premier ensemble de propriétés a pour but de décrire la situation réelle de coopération qu'il s'agira par la suite de visualiser. Le champ d'application du modèle IVF se limite aux techniques de visualisation des interactions se produisant dans une situation de coopération. Deux types d'objets fondamentaux sont considérés dans cette optique : les *entités* qui interagissent entre elles et les *interactions* entre les entités. Nous identifierons donc des propriétés associées aux entités et d'autres associées aux interactions.

3.1.1. Propriétés descriptives des entités réelles

a. *Entités en interaction*

La première propriété, nommée '*Entités en interactions*', permet de spécifier la nature des objets en interactions. Deux types d'entités seront considérés : les personnes et les ressources.

Notons que cette propriété n'est pas '*à valeur unique*' car il est évident qu'une technique de visualisation peut représenter simultanément des personnes et des ressources.

Entités en interaction : {personnes, ressources}

Exemples :

La visualisation de conversation par messages électroniques '*Thread Arcs*' se verra affectée la valeur '*personnes*' (cf. Fig. 14, p. 88). C'est également le cas de la technique '*Chat Circles*' (cf. Fig. 28, p. 96).

La technique de visualisation '*VCV*' prendra les valeurs '*personnes*' et '*ressources*' car elle représente à la fois des échanges entre personnes par *chat* et vidéoconférence et des actions sur un *notebook* qui est une ressource (cf. Fig. 65, p. 120).

b. *Localisation des entités*

La propriété '*Localisation des entités*' précise si les entités sont localisées au même endroit ou non pendant leurs interactions. Dans le premier cas, la valeur '*co-localisées*' sera affectée. Dans le second, la valeur '*distantes*' sera fixée. Cette propriété fait référence à un des axes de la matrice de Johanssen [99] visant à classifier les situations de travail coopératif.

Cette propriété est multi-valuée car une même technique de visualisation peut représenter des interactions entre des personnes distantes et d'autres localisées au même endroit.

Localisation des entités : {co-localisées, distantes}

Exemple :

Les représentations des interactions se produisant au cours d'un projet de construction proposées par Howard et Petersen (2001) illustrent parfaitement cette propriété (cf. Fig. 68). Dans leurs sociogrammes, ils représentent, selon les cas, des interactions entre des entités distantes (p.ex. courriels) et/ou des interactions entre des entités localisées au même endroit (p.ex. réunions).

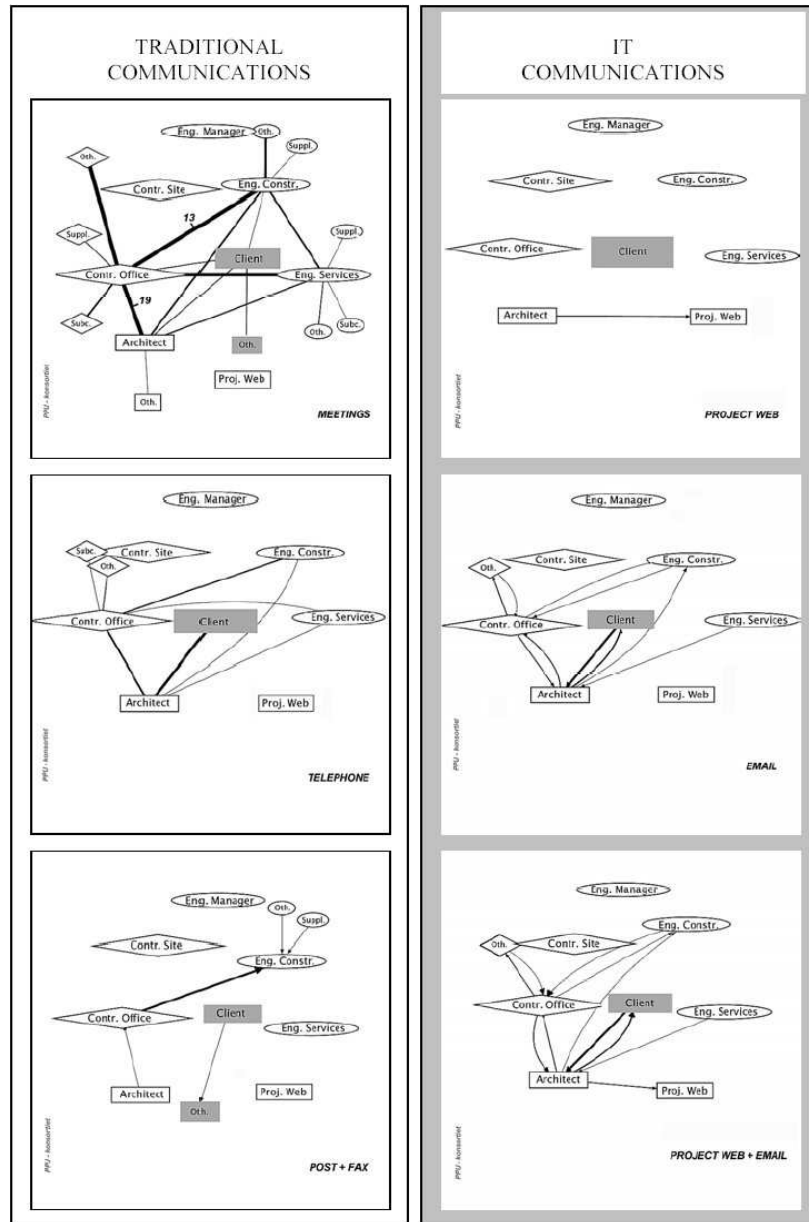


Fig. 68: Représentations des diverses interactions au cours d'un projet de construction (source : [94])

3.1.2. Propriétés descriptives des interactions réelles

a. Mode d'interaction

La propriété '**Mode d'interaction**' spécifie si les interactions concernées se produisent en mode synchrone ou asynchrone.

Dans le cas des interactions '*interpersonnelles*', elle fait référence au caractère instantané ou différé des échanges d'information. Par exemple, des communications par *chat* sont synchrones alors que des courriels sont asynchrones.

Pour les actions sur des ressources, cette propriété indique si les entités agissent simultanément sur ces ressources (*synchrone*) ou de manière séquentielle (*asynchrone*). Par exemple, un système de co-conception qui permet aux ingénieurs d'interagir en même temps sur un modèle numérique unique sera qualifié de *synchrone* alors qu'un système de gestion de documents qui supporte le versionnage sera qualifié d'*asynchrone*.

Cette propriété est multi-valuée car une technique de visualisation peut représenter à la fois des interactions en modes synchrone et asynchrone.

Mode d'interaction : {synchrone, asynchrone}

b. Cardinalité des interactions

La propriété '*Cardinalité des interactions*' spécifie le nombre d'entités qui peuvent prendre potentiellement part aux interactions concernées. Deux valeurs sont possibles, selon que les interactions se produisent entre deux entités individuelles (*dyadique*) ou entre plusieurs entités (*multiple*).

Cette propriété est multi-valuée car une technique de visualisation peut représenter simultanément les deux types d'interactions.

Cardinalité des interactions : {dyadiques, multiples}

Exemples :

La visualisation '*Themail*' est qualifiée de '*dyadique*' car elle ne représente que les messages électroniques échangés entre l'utilisateur courant et un de ses contacts (cf. Fig. 22, p. 92). Les visualisations '*Loom*' sont '*multiples*' car elles représentent des conversations dans des groupes de discussion auxquelles plusieurs personnes prennent part (cf. Fig. 33, p. 99).

3.2 Propriétés descriptives du modèle du monde réel

La représentation graphique d'une situation réelle de coopération nécessite de la modéliser sous une certaine forme. Le but de cette étape de modélisation est de sélectionner, parmi la multitude d'éléments composant le monde réel, ceux qui seront à visualiser.

Un ensemble de propriétés ont été intégrées dans la taxonomie IVF pour décrire les composantes du modèle retenu. Les propriétés ont été logiquement regroupées en deux ensembles selon qu'elles concernent les *entités* ou les *interactions*.

3.2.1. Propriétés descriptives des entités du modèle

a. Objets visualisables associés aux entités

La propriété '*Objets visualisables associés aux entités*' spécifie si le jeu de données disponible limite la technique de visualisation à afficher seulement des *entités* ou y ajoute des *attributs de ces entités*.

Notons dans ce contexte que nous ne considérerons pas l'identifiant des entités comme une propriété, bien qu'il puisse être considéré comme tel, d'un point de vue rigoureux. Cette restriction a été introduite pour garantir le caractère discriminant de la propriété (cf. p. 83). En effet, si on considère l'identifiant des entités comme une propriété, la quasi-totalité des techniques visualisant des entités se verraient affecter les deux valeurs.

Cette propriété est multi-valuée car une technique de visualisation peut disposer des données nécessaires pour représenter simultanément des entités en interaction et certains de leurs attributs.

Objets visualisables associés aux entités : {entités, attributs des entités}

Exemple :

La visualisation des personnes en interaction en ne renseignant que leur nom prendra la valeur '*entités*' (cf. Fig. 69 a) alors que si on représente également leur position hiérarchique la valeur '*attributs des entités*' sera aussi affectée (cf. Fig. 69 b).

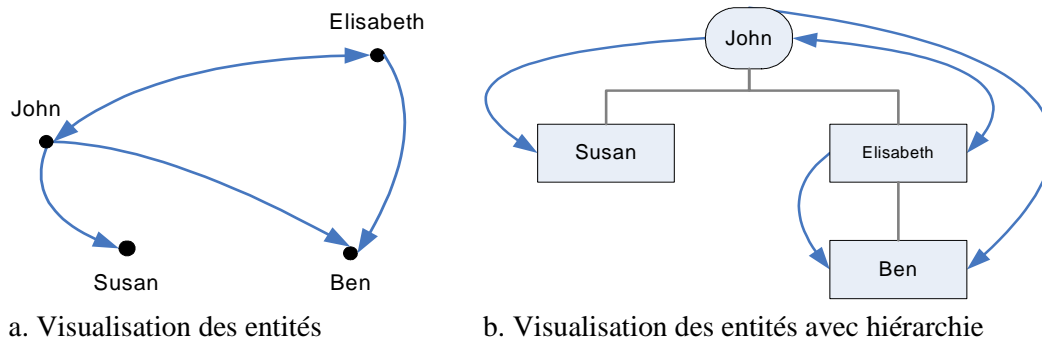


Fig. 69: Visualisation des entités

b. Positionnement temporel des entités

La propriété '*Positionnement temporel des entités*' spécifie si la technique de visualisation inclut une représentation du positionnement temporel d'événements associés aux entités. Cette représentation peut prendre une forme textuelle, graphique ou autre. Trois valeurs sont possibles, selon que la visualisation ne représente pas cette information (*absent*), qu'elle représente seulement la séquence d'événements (*relatif*) ou qu'elle représente le moment d'occurrence absolu des événements (*absolu*).

Cette propriété est multi-valuée car une même technique de visualisation peut proposer de visualiser la position temporelle de certains événements de manière absolue et celle d'autres de manière relative.

Il faut noter que cette propriété ne concerne pas les événements associés aux interactions. La représentation temporelle de ceux-ci fera l'objet d'une autre propriété (cf. **Positionnement temporel des interactions**).

Positionnement temporel des entités : {absent – {relatif, absolu}}

Exemple :

Dans la technique de visualisation '*Fugue*', les intervenants prenant part à une conversation par *chat* sont positionnés verticalement selon l'ordre dans lequel ils ont rejoint la conversation (cf. Fig. 30, p. 97). Ces événements associés aux entités ne sont cependant pas représentés de manière à visualiser leur position absolue dans le temps (date, heure...). La valeur '*relatif*' sera donc affectée.

c. Granularité des entités

La propriété '**Granularité des entités**' spécifie le niveau d'aggrégation des entités représentées par la technique de visualisation. On distinguera deux valeurs selon que celle-ci concerne des entités individuelles ou des groupes d'entités (*entités agrégées*).

Cette propriété est multi-valuée car une technique de visualisation peut représenter simultanément des entités et des groupes d'entités.

Granularité des entités : {entités individuelles, entités agrégées}

Exemples :

La technique de visualisation '*Babble*' illustre un cas où des entités individuelles sont affichées. Il s'agit d'une visualisation originale de conversations par *chat* entre des individus représentés par des cercles colorés. La valeur '*entités individuelles*' s'applique donc à cette technique. Par contre, Perry (1997) propose une représentation des interactions se produisant au cours de projets de construction (cf. Fig. 70). Dans sa représentation, certaines entités sont représentées sous une forme individuelle (p.ex. *Resident Engineer*) et d'autres sous forme agrégée (p.ex. *Construction Team*). Nous affecterons donc les deux valeurs '*entités individuelles*' et '*entités agrégées*'.

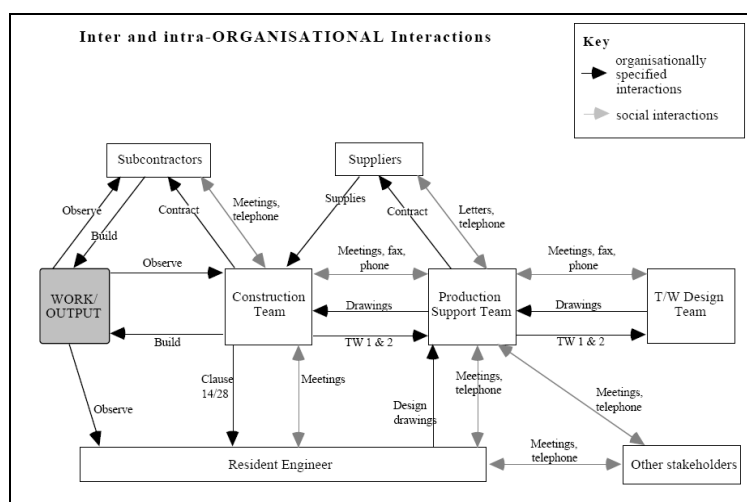


Fig. 70: Visualisation des interactions au cours d'un projet (source : [152])

3.2.2. Propriétés descriptives des interactions du modèle

a. Objets visualisables associés aux interactions

La propriété '*Objets visualisables associés aux interactions*' spécifie si le jeu de données disponible limite la technique de visualisation à afficher seulement des *interactions* ou inclut aussi des *attributs de ces interactions*.

Cette propriété est multi-valuée car une technique de visualisation peut disposer des données nécessaires pour représenter simultanément des interactions et certaines propriétés de celles-ci.

Objets visualisables associés aux interactions : {*interactions, attributs des interactions*}

Exemple :

La visualisation de calendrier proposée par Mynatt et Tullio (2001) représente à la fois des interactions (les 'événements' stockés dans un agenda) et certains de leurs attributs, comme la probabilité d'y assister (cf. Fig. 61, p. 117).

b. Positionnement temporel des interactions

La propriété '*Positionnement temporel des interactions*' spécifie si la technique de visualisation représente le positionnement temporel d'événements associés aux interactions. Cette représentation peut prendre une forme textuelle, graphique ou autre. Trois valeurs sont permises, selon que la visualisation ne représente pas cette information (*absent*), qu'elle représente seulement la séquence des interactions (*relatif*) ou qu'elle représente le moment d'occurrence absolu de celles-ci (*absolu*).

Cette propriété est multi-valuée car une même technique de visualisation peut proposer de visualiser la position temporelle de certains événements de manière absolue et celle d'autres de manière relative.

Positionnement temporel des interactions : {*absent – {relatif, absolu}*}

Exemples :

La technique de visualisation '*Fugue*' représente la séquence des messages (c.-à-d. des interactions dans le modèle IVF) postés dans un environnement de communication basé sur le *chat*. La taille des caractères avec lesquels les messages sont affichés décroît à mesure que le temps s'écoule depuis le moment où le message a été posté (cf. Fig. 30, p. 97). La taille de caractère normale correspond à des messages en train d'être postés. En comparant la taille des caractères, on peut donc reconstituer la séquence des messages. On assignera donc la valeur '*relatif*'. Il faut remarquer que la direction verticale n'est pas associée au positionnement temporel des messages mais à celui des entités (cf. description de cette propriété).

Dans une des options de la technique de visualisation '*History Flow*', les actions réalisées sur des pages web d'un site Wikipedia sont visualisées sur une ligne du temps, ce qui permet de les positionner temporellement de manière absolue (cf. Fig. 48 b, p. 109). La valeur '*absolu*' sera donc justifiée.

La technique '*VisuThread*' visualise le moment d'occurrence de messages électroniques de manière absolue pour les messages déjà envoyés et relative pour les messages restant à envoyer (cf. Fig. 15, p. 88). Elle doit donc recevoir les valeurs '*absolu*' et '*relatif*'.

c. *Granularité des interactions*

La propriété '*Granularité des interactions*' spécifie le niveau d'aggrégation des interactions représentées par la technique de visualisation. On distinguera deux valeurs selon que celle-ci permet de visualiser des interactions individuelles ou des ensembles d'interactions (*interactions agrégées*).

Cette propriété est multi-valuée car une technique de visualisation peut représenter simultanément des interactions et des groupes d'interactions.

Granularité des interactions : {interactions individuelles, interactions agrégées}

Exemple :

La technique '*Correspondent map*' visualise les messages envoyés à chaque correspondant sous la forme d'un rectangle dont la taille est proportionnelle à leur nombre (cf. Fig. 25, p. 93). Elle reçoit donc la valeur '*interactions agrégées*'.

d. *Centricité des interactions*

La propriété '*Centricité des interactions*' spécifie si les interactions sont relatives à une entité particulière (*égocentrique*) ou si elles concernent un ensemble d'entités sans en favoriser une en particulier (*sociocentrique*).

Cette propriété est uni-valuée.

Centricité des interactions : {égocentrique – sociocentrique}

Exemples :

La visualisation '*CrystalChat*' est *égocentrique* car elle représente les messages échangés entre une personne de référence et ses interlocuteurs (cf. Fig. 36, p. 100). La technique '*PeopleGarden*' est *sociocentrique* car elle visualise la participation de tous les membres d'un groupe à un tableau de messages (cf. Fig. 39, p. 102).

e. *Temporalité des interactions*

La propriété '*Temporalité des interactions*' fait référence à quelles interactions sont représentées par rapport au temps présent. Trois valeurs sont permises : *interactions du passé*, *interactions en cours*, *interactions futures n'ayant pas encore eu lieu*.

Deux remarques sont nécessaires dans ce contexte.

La première concerne les interactions en cours. Vu que le temps présent est par définition instantané et donc de durée infiniment courte, on considèrera que la technique visualise des interactions en cours lorsqu'elle limitera, par un moyen quelconque, la période de temps concernée à un intervalle de passé récent.

La seconde est relative aux interactions futures. Par définition, et contrairement aux interactions passées ou en cours, les interactions futures sont hypothétiques. Dès lors, on associera à la valeur '*interactions futures*' des interactions planifiées ou probables (p.ex. afficher les prochaines réunions prévues).

Cette propriété est multi-valuée car une technique de visualisation peut représenter simultanément des interactions passées, en cours et futures.

Temporalité des interactions :
{interactions passées, interactions en cours, interactions futures}

Exemples :

La technique de visualisation '*Conversation Landscape*' est dédiée à la représentation d'archives de conversations par *chat*. Les personnes sont représentées par des lignes verticales colorées et les messages par des lignes horizontales dont la longueur est proportionnelle à leur taille (cf. Fig. 71). L'axe vertical est associé au temps. Cette technique de visualisation se verra affecter la valeur '*interactions passées*'.

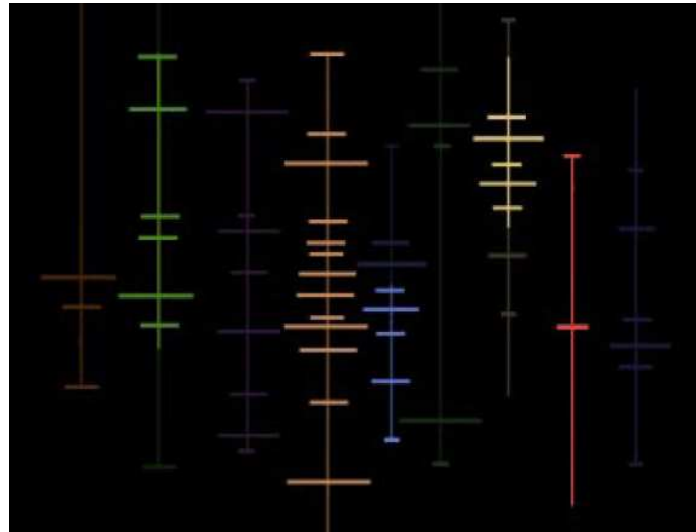


Fig. 71: Technique de visualisation: '*Conversation Landscape*', source : [54]

La technique de visualisation '*VisuThread*' permet de représenter des conversations basées sur des messages électroniques (cf. Fig. 15, p. 88). Elle repose sur la représentation d'interactions passées, en cours mais aussi futures. Dans ce cas, les interactions futures sont associées aux messages contenus dans le répertoire '*draft*' du client de messagerie. Elles sont représentées par des flèches en ligne pointillée. Les valeurs '*interactions passées*', '*interactions en cours*' et '*interactions futures*' seront donc affectées toutes les trois.

f. Durée des interactions

La propriété '*Durée des interactions*' indique si les interactions concernées sont instantanées et/ou se produisent pendant un intervalle de temps d'une durée perceptible. Deux valeurs sont permises : *interactions instantanées* et *interactions prolongées ayant une certaine durée*.

Durée des interactions : {instantanées, prolongées}

Exemple :

Les visualisations d'agenda telles que '*Availability Bars*' (cf. Fig. 60, p. 116) représentent essentiellement des réunions et des rendez-vous qui sont des interactions *prolongées*. Dans certains cas, elles peuvent également inclure des visualisations d'événements *instantanés* (p.ex. un rappel pour effectuer une tâche). De même, le système '*Task View*' (cf. Fig. 26, p. 94) combine dans une même vue des interactions *instantanées* (courriels) et *prolongées* (tâches).

Par contre, un système de visualisation tel que '*Thread Arcs*' visualise des courriels qui sont, par définition, instantanés.

3.3 Propriétés descriptives de la technique de visualisation

Le troisième grand ensemble de propriétés a pour but de qualifier la technique de visualisation proprement dite. A ce niveau, la taxonomie IVF choisit de distinguer les techniques de visualisation par le biais de propriétés génériques. Elle ne regroupe donc pas les techniques rencontrées en types tels que ‘graphe’, ‘histogramme’ ou ‘arbre’.

Cette démarche est basée sur notre étude des taxonomies. Elle nous a incités à concevoir la taxonomie IVF de manière telle qu’elle puisse s’appliquer à la majorité des nouvelles techniques de visualisation d’interactions sans nécessiter l’ajout de nouvelles valeurs pour les différentes propriétés. Le cas échéant, si une nouvelle technique de visualisation révolutionnaire était proposée, la taxonomie IVF s’adapterait en ajoutant une nouvelle valeur ou propriété générique plutôt qu’en créant un nouveau type de visualisation.

Les propriétés associées à la description de la technique de visualisation sont groupées en différents sous-ensembles : celles qui qualifient la technique en général, celles qui décrivent le contexte d’utilisation de la visualisation et celles qui spécifient comment les entités et les interactions sont visualisées.

3.3.1. Propriétés descriptives de la technique de visualisation en général

a. *Dimensions de l’espace de visualisation*

La propriété ‘*Dimensions de l’espace de visualisation*’ est définie comme le nombre de coordonnées spatiales nécessaires pour positionner les données dans l’espace de visualisation. Trois valeurs sont possibles : représentations à une, deux ou trois dimensions.

Il faut noter que cette propriété ne fait pas référence à la dimensionnalité des données.

Notons que certains auteurs proposent des représentations qu’ils définissent comme possédant quatre dimensions. Il s’agit en fait de représentations de l’évolution au cours du temps de structures tri-dimensionnelles. Pour notre part, nous considérons ces techniques comme tri-dimensionnelles et nous utiliserons d’autres propriétés pour décrire la composante temporelle.

Cette propriété est uni-valuée.

Dimensions de l’espace de visualisation : {1 dimension – 2 dimensions – 3 dimensions}

Exemples :

La visualisation d’agenda ‘*DateLens*’ (cf. Fig. 53, p. 113) repose sur l’utilisation d’une matrice à *deux dimensions*. Le système ‘*Archichronos*’ (cf. Fig. 42, p. 105) visualise les actions sur un modèle numérique en *trois dimensions*.

b. *Variabilité de l’espace d’affichage*

La propriété ‘*Variabilité de l’espace d’affichage*’ définit la relation entre le volume des données à visualiser et la taille de l’espace d’affichage. Elle spécifie si la technique de visualisation utilise un espace d’affichage de taille fixe indépendamment du volume de données. On associera la valeur ‘*fixe*’ aux techniques qui utilisent un espace de visualisation constant, quel que soit le nombre de données à représenter (cf. *space-filling methods*). On affectera la valeur ‘*variable*’ lorsque l’espace nécessaire à la visualisation augmente (ou devrait augmenter) avec le volume de données.

On peut argumenter que toute technique pourrait recevoir la valeur ‘*fixe*’ car il suffit d’augmenter la résolution de la représentation pour augmenter le nombre de données représentables. Cet argument n’est cependant pas acceptable en pratique. En effet, d’une part, la résolution effective des appareils de visualisation n’est pas illimitée et d’autre part, l’augmentation pure et simple de la densité d’information conduit à des représentations qui

réduisent fortement, voire interdisent leur compréhension. Par exemple, dans le cas de représentations sous la forme de graphes nœuds-liens, l'augmentation du nombre de nœuds ou de liens dans un espace d'affichage fixe conduit, à partir d'une certaine limite, à une représentation inutilisable. On associera dès lors à ce type de visualisation la valeur '*variable*'.

Cette propriété est uni-valuée.

Variabilité de l'espace d'affichage : {fixe – variable}

Exemple :

Les '*smartmaps*' (cf. Fig. 44, p. 106) visualisent les opérations sur des ressources. Les *smartmaps* sont une variante des *treemaps*. Ils sont des représentations dont la taille est fixée, indépendamment du nombre de données à représenter. La valeur '*fixe*' sera donc appropriée.

Par contre, le système '*Crystal Chat*' (cf. Fig. 36, p. 100) est qualifié de '*variable*' car si le nombre de messages composant une conversation augmente la ligne de points associés augmente au même rythme et nécessite donc un espace d'affichage croissant. Conscients de la problématique, ses concepteurs ont d'ailleurs inclus des fonctionnalités de zoom et de filtrage.

c. Distorsion géométrique

La propriété '*Distorsion géométrique*' spécifie si la technique de visualisation utilise une fonction unique associant les données à leur position dans la représentation graphique.

Dans une vue déformée, les valeurs associées aux données ne sont pas toutes mises en correspondance de manière équivalente avec leur représentation géométrique dans l'espace d'affichage. Cette propriété fait notamment référence aux travaux de Furnas [67].

Cette propriété est uni-valuée.

Distorsion géométrique : {déformé – non déformé}

Exemples :

La visualisation '*DateLens*' (cf. Fig. 53, p. 113) utilise la technique de déformation en œil de poisson pour représenter un agenda. Elle se verra donc affecter la valeur '*déformé*'. En effet, les données magnifiées ne sont pas représentées avec la même loi que les autres.

Par contre, malgré son caractère moins habituel, une représentation en arbre circulaire hyperbolique n'est pas déformée car toutes les données sont positionnées dans l'espace d'affichage avec la même loi (hyperbolique).

3.3.2. Propriétés descriptives du contexte d'utilisation de la visualisation

Le troisième ensemble de propriétés a pour but de préciser dans quel contexte la visualisation est utilisée. Une fois encore, l'approche suivie par le modèle IVF consiste à proposer des propriétés génériques. Nous ne proposerons donc pas de propriétés visant, par exemple, à préciser un secteur d'activité, une profession ou la tranche d'âge des utilisateurs.

a. Rôle de l'utilisateur

La propriété '*Rôle de l'utilisateur*' spécifie si la technique de visualisation est destinée à être utilisée par un acteur qui prend part aux interactions ou par un observateur extérieur.

Cette propriété est multi-valuée.

Rôle de l'utilisateur : {acteur participant, observateur externe}

Exemples :

Le système ‘*Babble*’ (cf. Fig. 29 p. 96) a pour utilisateurs les participants à une conversation via messagerie instantanée. Elle reçoit donc la valeur ‘*acteur participant*’.

La visualisation ‘*CodeSaw*’ (cf. Fig. 46, p. 108) montre les participations respectives de plusieurs programmeurs au code d’un programme. Elle peut être destinée à un *acteur participant* mais aussi à un *observateur externe* (p.ex. le responsable du projet). Elle prendra donc les deux valeurs.

b. But de la visualisation

La propriété ‘***But de la visualisation***’ spécifie si la technique de visualisation est destinée à supporter une entité pour interagir avec d’autres entités ou si elle a pour but d’analyser des interactions.

Cette propriété est multi-valuée.

But de la visualisation : {support aux interactions, analyse des interactions}

Exemples :

Le monde virtuel 3-D du système ‘*Palace*’ (cf. Fig. 31, p. 98) a pour but de *supporter les interactions* par messagerie instantanée. Par contre, la visualisation ‘*Newsgroup Crowds*’ (cf. Fig. 38, p. 102) représente l’historique d’interactions dans un groupe pour identifier les acteurs principaux. Elle est conçue pour l’*analyse des interactions*.

Il faut remarquer que, bien que proche de la précédente, cette propriété est néanmoins différente. A titre d’exemple, une technique de visualisation peut être utilisée par un acteur participant à des interactions à la fois à des fins de support et à des fins d’analyse. Par contre, il est difficile d’imaginer une technique qui serait utilisée par un observateur externe pour supporter des interactions puisqu’il ne participe pas à celles-ci.

c. Intégration

La propriété ‘***Intégration***’ spécifie si la technique de visualisation est intégrée ou indépendante de l’outil informatique utilisé pour instancier les interactions. Cette distinction influence directement le contexte dans lequel la visualisation peut être utilisée.

Cette propriété est uni-valuée.

Intégration : {intégré – indépendant}

Exemples :

La visualisation ‘*Fugue*’ (cf. Fig. 30, p. 97) est *intégrée* dans l’outil de messagerie instantanée. La visualisation ‘*Correspondent Tremap*’ (cf. Fig. 25, p. 93) se base sur un fichier qui résulte d’un traitement opéré sur des archives de messagerie électronique. Elle est *indépendante* du système de messagerie proprement dit.

3.3.3. Propriétés descriptives de la représentation des entités

a. Représentation graphique des entités

La propriété '**Représentation graphique des entités**' spécifie quelle technique est employée pour représenter les entités en interaction. Plusieurs approches sont possibles : ne pas représenter les entités, représenter les entités avec du texte, ou utiliser des objets graphiques (point, ligne, objet surfacique, objet volumique).

Cette propriété est multi-valuée.

Représentation graphique des entités :

{*aucun, texte, objet ponctuel, objet linéaire, objet surfacique, objet volumique*}

Exemples :

Dans la visualisation '*CodeSaw*' (cf. Fig. 46, p. 108), les personnes sont représentées par des lignes horizontales. La valeur '*objet linéaire*' est donc affectée.

Dans le système '*Tower World*' (cf. Fig. 51, p. 111), les entités sont représentées par des objets 3-D dans un monde virtuel. Il reçoit donc la valeur '*objet volumique*'.

Enfin, une technique peut visualiser des interactions dans lesquelles des personnes sont impliquées sans visualiser ces dernières. C'est par exemple le cas d'un histogramme qui indique le nombre de visites quotidiennes sur un site web. Les personnes en interaction (c.-à-d. celles qui visitent le site) ne sont pas représentées. La valeur '*aucun*' doit alors être mentionnée pour la propriété.

b. Positionnement des entités dans l'espace d'affichage

La propriété '**Positionnement des entités dans l'espace d'affichage**' spécifie si la position spatiale (à une, deux ou trois coordonnées) des objets graphiques associés aux entités est sémantiquement significative.

Cette propriété fait référence aux variables du plan de Bertin (cf. p. 39) ou aux variables associées à la position dans la taxonomie de Card et Mackinlay (cf. p. 53). Nous n'avons pas adopté leur formalisme (x,y,z) car il fait trop référence intuitivement aux systèmes de coordonnées orthogonaux. Comment considérer dans leur cas une représentation utilisant un système de coordonnées polaires (r, θ) ? Dans notre approche, ce qui est important de distinguer, c'est le fait que la position des objets graphiques soit porteuse de sens ou non.

Cette propriété est uni-valuée.

Positionnement des entités dans l'espace d'affichage :

{*position significative – position non significative*}

Exemples :

Dans la visualisation '*Babble*' (cf. Fig. 29, p. 96), les personnes intervenant dans la conversation sont représentées par des cercles dont la position indique le niveau de participation. La position est donc *significative*. Dans la visualisation '*CrystalChat*' (cf. Fig. 36, p. 100), les personnes sont associées à des demi-plans distribués radialement de manière homogène. La position des demi-plans n'est pas *significative*.

c. Variables rétinienne associées aux entités

La propriété '**Variables rétinienne associées aux entités**' spécifie quelles variables rétinienne sont utilisées pour représenter les entités et leurs propriétés.

Cette propriété se base essentiellement sur le travail de Bertin en sémiologie graphique.

Cette propriété est multi-valuée.

Variables rétinienne associées aux entités :

{couleur, niveau de gris, forme, orientation, taille, grain, texture}

Exemples :

La visualisation '*Chat Circles*' (cf. Fig. 28, p. 96) associe chaque personne à un cercle coloré. La valeur '*couleur*' doit donc être affectée.

La technique '*Clique Panel*' (cf. Fig. 20, p. 91) associe les groupes de personnes qui participent à des communications communes (c.-à-d. les cliques) à des nœuds positionnés sur le périmètre d'un cercle. Les nœuds sont représentés graphiquement par des polygones dont le nombre de côtés indique le nombre de personnes incluses dans la clique. Il faut donc lui affecter la valeur '*forme*'.

d. Niveau d'abstraction associé aux entités

La propriété '**Niveau d'abstraction associé aux entités**' spécifie si les objets graphiques utilisés pour représenter les entités sont des formes purement abstraites, rappellent des objets réels via des métaphores ou miment la réalité (*pseudo-réel*).

Cette propriété est uni-valuée.

Niveau d'abstraction associé aux entités : *{abstrait – métaphorique – pseudo-réel}*

Exemples :

La vue '*Network View*' du système '*Soylent*' (cf. Fig. 19, p. 90) associe les personnes aux nœuds d'un graphe. Elle reçoit donc la valeur '*abstrait*'. Le système '*Tower World*' (cf. Fig. 51, p. 111) visualise les personnes par des avatars. Nous lui assignerons donc la valeur '*pseudo-réel*'.

e. Animation des entités

La propriété '**Animation des entités**' spécifie si du mouvement ou de l'animation sont associés à la représentation graphique des entités.

Cette propriété est uni-valuée.

Animation des entités : *{statique – animé}*

Exemple :

Dans la visualisation '*Babble*' (cf. Fig. 29, p. 96), les personnes sont associées à des cercles de couleur. Ces cercles se déplacent vers le centre d'un disque quand les personnes sont actives dans la conversation et s'en éloignent quand elles arrêtent d'intervenir. Elle se voit donc affecter la valeur '*animé*'.

f. Interactivité associée aux entités

La propriété '***Interactivité associée aux entités***' fait référence à la présence ou à l'absence de moyens pour l'utilisateur d'interagir avec la représentation graphique des entités.

Nous considérerons qu'une technique de visualisation est interactive si l'utilisateur peut dynamiquement modifier la sélection des entités à afficher, les propriétés affichées pour ces entités ou un paramètre de la sortie graphique utilisée pour leur affichage.

Cette propriété est multi-valuée.

***Interactivité associée aux entités : {sélection des entités,
sélection des propriétés des entités à afficher, sélection du rendu graphique des entités}***

Exemple :

Dans l'agenda de groupe de *Lotus Notes* (cf. Fig. 55, p. 114), l'utilisateur peut choisir les personnes dont il souhaite voir l'agenda. Il existe donc une fonction de '*sélection des entités*'.

Dans le système '*Chat circle*' (cf. Fig. 28, p. 96), les participants à la conversation peuvent choisir la couleur du cercle qui les représente, ce qui correspond à la valeur '*sélection du rendu graphique des entités*'.

3.3.4. Propriétés descriptives de la représentation des interactions

a. Représentation graphique des interactions

La propriété '***Représentation graphique des interactions***' spécifie quel type d'objet graphique est utilisé pour représenter les interactions. Plusieurs approches sont possibles : ne pas représenter les interactions, représenter les interactions sous la forme de texte, ou utiliser des objets graphiques (point, ligne, objet surfacique, objet volumique).

Cette propriété est multi-valuée.

***Représentation graphique des interactions :
{aucun, texte, objet ponctuel, objet linéaire, objet surfacique, objet volumique}***

Exemples :

Dans le système '*Comic Chat*' (cf. Fig. 32, p. 98), les interactions, c.-à-d. les messages échangés, sont représentées sous forme *textuelle*. Par contre, dans le système '*Fugue*', elles sont visualisées par une combinaison de *texte* et d'un *objet linéaire* (cf. Fig. 37, p. 101).

b. Positionnement des interactions dans l'espace d'affichage

La propriété '***Positionnement des interactions dans l'espace d'affichage***' spécifie si la position spatiale (à une, deux ou trois coordonnées) des objets graphiques associés aux interactions est sémantiquement significative.

Cette propriété est uni-valuée.

***Positionnement des interactions dans l'espace d'affichage :
{position significative – position non significative}***

Exemple :

Dans la technique '*Authorlines*' (cf. Fig. 40, p. 103), les conversations sont représentées par des cercles qui sont placés au-dessus ou au-dessous d'une ligne du temps selon qu'elles aient été initiées par l'utilisateur de référence ou non. La position est donc *significative*.

c. Variables rétinienne associées aux interactions

La propriété '**Variables rétinienne associées aux interactions**' spécifie quelles variables rétinienne sont utilisées pour représenter les interactions et leurs propriétés.

Cette propriété se base sur le travail de Bertin en sémiologie graphique.

Cette propriété est multi-valuée.

Variables rétinienne associées aux interactions :
{couleur, niveau de gris, forme, orientation, taille, grain, texture}

Exemples :

Dans le système '*Archichronos*' (cf. Fig. 42, p. 105), les interactions sont les modifications apportées au modèle numérique du bâtiment. Elles sont visualisées par le biais de *couleurs*. Dans la technique '*Authorlines*' (cf. Fig. 40, p. 103), les interactions sont les conversations qui sont visualisées par des cercles *colorés* dont la *taille* est proportionnelle au nombre de messages qu'elles incluent.

d. Niveau d'abstraction associé aux interactions

La propriété '**Niveau d'abstraction associés aux interactions**' spécifie si les objets graphiques utilisés pour représenter les interactions sont des formes purement abstraites, rappellent des objets réels via des métaphores ou miment la réalité (*pseudo-réel*).

Cette propriété est uni-valuée.

Niveau d'Abstraction associé aux interactions :{abstrait - métaphorique – pseudo-réel}

Exemples :

La technique de visualisation '*People Garden*' utilise la métaphore d'un jardin rempli de fleurs pour représenter des interactions basées sur du *chat* (cf. Fig. 39, p. 102). Elle reçoit donc la valeur '*métaphorique*'. La technique de visualisation '*VCV*' représente les interactions par des glyphes abstraits (cf. Fig. 65, p. 120). Il faut donc lui assigner la valeur '*abstrait*'.

e. Animation des interactions

La propriété '**Animation des interactions**' spécifie si du mouvement ou de l'animation sont associés à la représentation graphique des entités.

Cette propriété est uni-valuée.

Animation des interactions :{statique – animé}

Exemple :

La visualisation '*Social Network Fragments*' inclut des animations qui montrent comment les relations sociales basées sur la messagerie électronique émergent et disparaissent au cours du temps (cf. Fig. 18, p. 90). Elle se voit donc assignée la valeur '*animé*'.

f. Interactivité associée aux interactions

La propriété '*Interactivité associée aux interactions*' fait référence à la présence ou à l'absence de moyens pour l'utilisateur d'interagir avec la représentation graphique des interactions.

Nous considérerons qu'une technique de visualisation est interactive si l'utilisateur peut dynamiquement modifier la sélection des interactions à afficher, les propriétés affichées pour ces interactions ou un paramètre de la sortie graphique utilisée pour leur affichage.

Cette propriété est multi-valuée.

Interactivité associée aux interactions : {sélection des interactions, sélection des propriétés des interactions à afficher, sélection du rendu graphique des interactions}

Exemple :

Le système '*Chat Circle*' (cf. Fig. 28, p. 96) permet de *sélectionner les interactions* visualisées par un moyen très original. Comme seules les interactions (c.-d. les messages) situés dans la périphérie du cercle associé à l'utilisateur courant sont représentées, on choisit celles-ci en déplaçant ce cercle dans un espace bi-dimensionnel.

3.4 Vue synoptique de la taxonomie IVF

La Table 10 propose une vue d'ensemble de la taxonomie IVF.

Table 10: Vue synoptique des propriétés de la taxonomie IVF

Propriétés descriptives de la situation réelle de coopération	
Propriétés descriptives des entités réelles	
<i>Entités en interaction :</i>	{personnes, ressources}
<i>Localisation des entités :</i>	{co-localisées, distantes}
Propriétés descriptives des interactions réelles	
<i>Mode d'interaction :</i>	{synchrone, asynchrone}
<i>Cardinalité des interactions :</i>	{dyadiques, multiples}
Propriétés descriptives du modèle du monde réel	
Propriétés descriptives des entités du modèle	
<i>Objets Visualisables associés aux Entités :</i>	{entités, attributs des entités}
<i>Positionnement Temporel des Entités :</i>	{absent – {relatif, absolu}}
<i>Granularité des Entités :</i>	{entités individuelles, entités agrégées}
Propriétés descriptives des interactions du modèle	
<i>Objets Visualisables associés aux Interactions :</i>	{interactions, attributs des interactions}
<i>Positionnement Temporel des Interactions :</i>	{absent – {relatif, absolu}}
<i>Granularité des Interactions :</i>	{interactions individuelles, interactions agrégées}
<i>Centricité des Interactions :</i>	{égocentrique – sociocentrique}
<i>Temporalité des Interactions :</i>	{interactions passées, interactions en cours, interactions futures}
<i>Durée des Interactions :</i>	{instantanées, prolongées}
Propriétés descriptives de la technique de visualisation	
Propriétés descriptives de la technique de visualisation en général	
<i>Dimensions de l'Espace de Visualisation :</i>	{1 dimension – 2 dimensions – 3 dimensions}
<i>Variabilité de l'Espace d’Affichage :</i>	{fixe – variable}
<i>Distorsion Géométrique :</i>	{déformé – non déformé}

Propriétés descriptives du contexte d'utilisation de la visualisation	
Rôle de l'utilisateur :	{acteur participant, observateur externe}
But de la visualisation :	{support aux interactions, analyse des interactions}
Intégration :	{intégré – indépendant}
Propriétés descriptives de la représentation des entités	
Représentation graphique des entités :	{aucun, texte, objet ponctuel, objet linéaire, objet surfacique, objet volumique}
Positionnement des entités dans l'espace d'affichage :	{position significative – position non significative}
Variables rétinienne associées aux entités :	{couleur, niveau de gris, forme, orientation, taille, grain, texture}
Niveau d'abstraction associé aux entités :	{abstrait – métaphorique – pseudo-réel}
Animation des entités :	{statique – animé}
Interactivité associée aux entités :	{sélection des entités, sélection des propriétés des entités à afficher, sélection du rendu graphique des entités}
Propriétés descriptives de la représentation des interactions	
Représentation graphique des interactions :	{aucun, texte, objet ponctuel, objet linéaire, objet surfacique, objet volumique}
Positionnement des interactions dans l'espace d'affichage :	{position significative – position non significative}
Variables rétinienne associées aux interactions :	{couleur, niveau de gris, forme, orientation, taille, grain, texture}
Niveau d'abstraction associé aux interactions :	{abstrait – métaphorique – pseudo-réel}
Animation des interactions :	{statique – animé}
Interactivité associée aux interactions :	{sélection des interactions, sélection des propriétés des interactions à afficher, sélection du rendu graphique des interactions}

Cette taxonomie nous a été utile tout au long de notre processus de conception de techniques de visualisation d'activités coopératives. Nous nous y sommes référés pour explorer de nouvelles pistes de travail et pour confronter nos idées avec des techniques existantes.

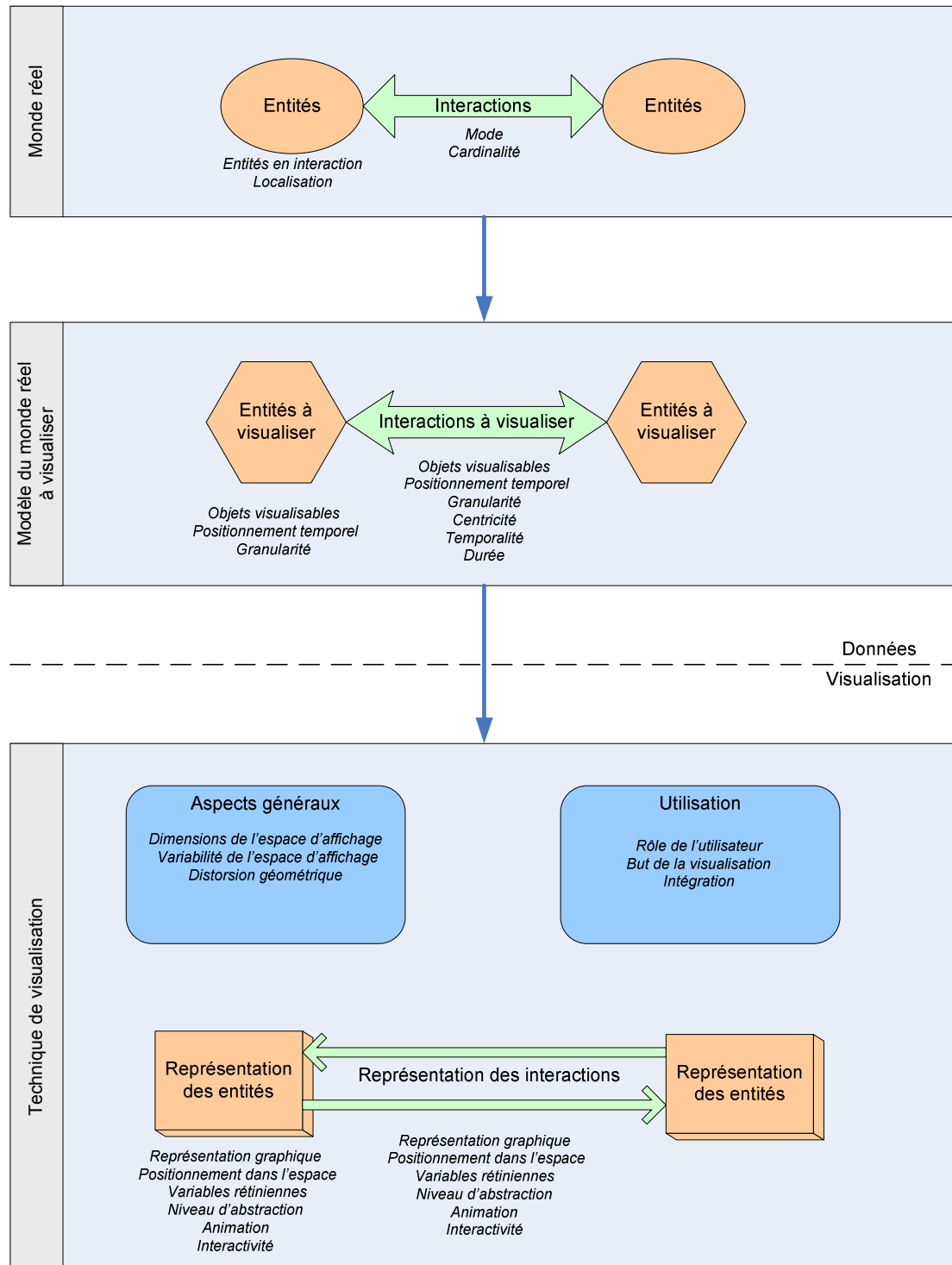


Fig. 72: Vue synoptique de la taxonomie IVF, avec les propriétés associées

Chapitre 6 : Indices d'activité et Interactions

« La seule vraie science est la connaissance des faits. »

G.L. Leclerc, comte de Buffon

1. Introduction

Nous avons précédemment étudié comment visualiser des informations relatives à des activités coopératives et nous avons proposé la taxonomie IVF qui permet de clarifier les concepts de ce domaine et les positionne les uns par rapport aux autres.

Dans ce chapitre, nous abordons de manière moins abstraite notre problématique en étudiant comment nous pouvons en pratique supporter la conscience de groupe dans un contexte coopératif. Pour ce faire, nous devons étudier en détail la nature des informations à communiquer et tenter de les classer sur base d'un modèle à la fois simple, pragmatique et représentatif. Nous essayerons aussi de concevoir des manières originales de les visualiser au moyen de logiciels.

2. Conscience de groupe

2.1 Travaux antérieurs

De nombreux travaux antérieurs dans le domaine des collecticiels ont montré l'importance de la notion de conscience de groupe. Dourish and Bellotti (1992) définissent ce concept comme *'une compréhension des activités des autres, ce qui fournit un contexte pour sa propre activité'*. Gutwin and Greenberg (1999) expliquent qu'*'il devient de plus en plus apparent qu'être capable de rester conscient des autres joue un rôle important dans la fluidité et le caractère naturel de la collaboration'*. La conscience de groupe peut prendre plusieurs formes : informelle, sociale, relative à la structure du groupe ou à l'espace de travail (Greenberg et al., 1996). Nous nous focaliserons dans ce chapitre sur la conscience de l'espace de travail (*workspace awareness*), qui a été définie comme *'la compréhension actualisée de l'interaction d'une autre personne avec l'espace de travail partagé. [...] Cela inclut la conscience de la présence de personnes, de comment elles interagissent avec l'espace de travail et des événements se produisant de celui-ci.'* (Gutwin et Greenberg, 2004)

Compte tenu de l'importance de la conscience de groupe, il est utile d'explorer quelles techniques peuvent être employées pour la supporter.

Diverses approches ont été développées, telles que :

- la notification des modifications via courriel,
- l'affichage du nombre de personnes en train de lire la même page que soi-même (Robinson et al., 2001),
- l'emploi d'icônes renseignant qui est en ligne (ex. IM Lotus Notes, cf. Fig. 49 a)
- l'utilisation d'avatars (cf. Fig. 31),
- la représentation des utilisateurs comme des formes abstraites (cf. Fig. 28),
- la représentation des actions des autres dans un environnement virtuel (Pinho et al., 2002).

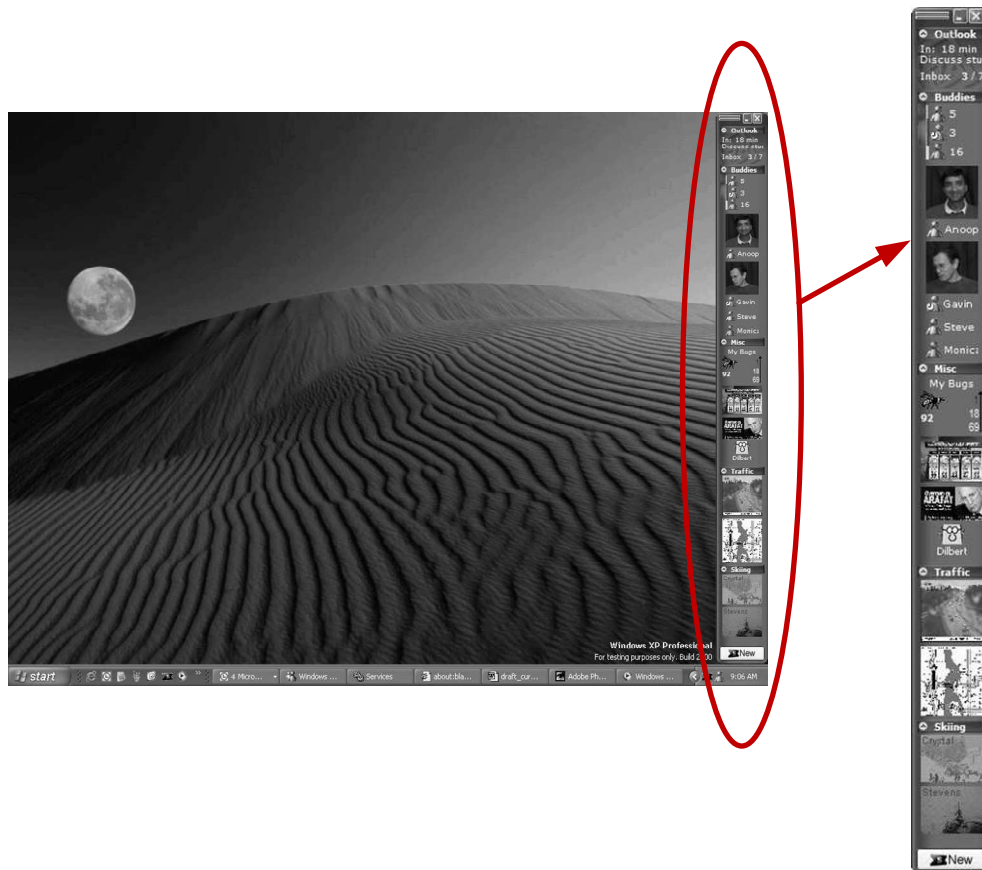


Fig. 73: Système Side Show pour supporter la conscience de groupe (source : [34])

On peut également remarquer que les indicateurs de conscience de groupe font dans certains cas partie intégrante de l'interface utilisateur du collecticiel (e.g. Fig. 44) et sont dans d'autres directement affichées dans l'interface liée au système d'exploitation (e.g. Cadiz et al., 2002 illustré par la Fig. 73).

2.2 Développements théoriques sur le concept

Pour rappel, nous avons choisi de modéliser la coopération comme un ensemble d'*interactions* entre des *entités* (cf. Fig. 10, p. 33).

Dans ce contexte, une *entité* peut être une personne ou une ressource (p.ex. un document, un epage web, un agenda de groupe, un objet partagé dans un environnement virtuel de conception...).

Une *interaction* est définie comme tout échange d'information entre des entités (p.ex. appel téléphonique, courriel, téléchargement d'un document à partir d'un site web, intervention sur un forum de discussion...). La création d'une interaction par une entité est appelée une *instanciation d'interaction*. Une interaction peut viser (c.-à-d. être destinée à) une ou plusieurs entités.

Cette définition englobe tous types d'interaction : électronique ou physique, synchrone ou asynchrone, co-localisée ou distante... Afin de limiter le concept à ce que nous étudions en pratique (les plates-formes électroniques de coopération), nous avons introduit le concept d'*interaction informatisée*, définie comme tout échange d'information entre des entités qui est basé sur l'utilisation d'outils informatiques.

Par ailleurs, l'utilisateur dont la conscience de groupe doit être supportée sera appelé l'*utilisateur de référence* (U_R). Comme nous l'avons montré dans la section précédente, les techniques pour supporter la conscience de groupe de U_R peuvent prendre plusieurs formes. Nous les avons catégorisées selon deux axes : l'*identification* et la *granularité*.

L'*identification* spécifie si l'information relative à la conscience de groupe qui est communiquée à U_R permet d'identifier les entités concernées. Deux cas sont possibles.

- L'information permet à U_R d'*identifier* les entités concernées. Cette approche repose le plus souvent sur la surveillance de l'ensemble ou d'une partie des entités de l'espace de travail et la notification à U_R des interactions qui le concernent. Les interactions concernent à la fois celles initiées par une entité surveillée (p.ex. informer U_R quand l'utilisateur U_A se connecte au système de *chat*) et celles qui visent une entité surveillée (p.ex. garder U_R informé des modifications apportées à un document partagé).
- La conscience de groupe *anonyme* fournit à U_R un feed-back sur ce qui se produit dans l'environnement collaboratif mais n'indique pas explicitement quelles entités sont impliquées. Par exemple, une icône spécifique est affichée quand une nouvelle (*news*) est déposée sur une plate-forme collaborative mais cette icône ne donne aucune information sur l'auteur de la nouvelle.

La *granularité* précise si l'information notifiée à U_R concerne des *entités spécifiques individuelles* au sein de l'espace de travail (p.ex. notification par courriel de la mise à jour d'un document D , icône associée au statut d'un utilisateur U_A) ou est *agrégée* (p.ex. affichage du nombre d'utilisateurs connectés à l'espace de travail mais pas de leur nom, affichage du nombre de documents chargés (*uploaded*) depuis la dernière visite de U_R mais sans fournir de détails sur les documents individuels).

Pour d'évidentes raisons de protection de la vie privée, de sécurité ou de surcharge informationnelle, seule une partie de toutes les interactions se produisant dans l'espace de travail sont en pratique notifiées à U_R . De plus, le sous-ensemble d'interactions notifiées diffère selon le type de conscience de groupe considéré. L'information '*identifiée*' ou '*individuelle*' est naturellement plus sensible que l'information '*anonyme*' ou '*agrégée*'. Par conséquent, le sous-ensemble d'informations notifiées agrégées ou anonymes peut être très large jusqu'à couvrir potentiellement toutes les interactions de l'espace de travail. Au contraire, l'ensemble des informations indentifiées et individuelles qui peut être communiquées à U_R est en principe beaucoup plus limité.

Comme l'indique la Table 11, quatre cas sont théoriquement possibles. Cependant, il apparaît que tous les cas ne présentent pas la même utilité en pratique. Communiquer de l'information identifiée et individuelle ou anonyme et agrégée est assez courant.

Table 11 : Catégorisation de l'information sur la conscience de groupe

Granularité	Identification	
	<i>identifié</i>	<i>anonyme</i>
<i>individuel</i>	- l'utilisateur U_A est en ligne - le document D a été modifié	- afficher le graphe des interactions sans identifier les nœuds
<i>agrégé</i>	- des membres de ce groupe d'utilisateurs ont envoyé des messages électroniques à U_R	- des utilisateurs sont connectés sur cette page web - des documents ont été ajoutés à l'espace de travail

Les deux autres cas sont plus rares. Une information anonyme et individuelle fait référence à la notification à U_R qu'une entité spécifique a été impliquée dans des interactions sans préciser de quelle entité il s'agit. Un autre exemple consiste à afficher les interactions entre certains utilisateurs comme un graphe 'nœuds-liens' sans identifier les nœuds (représentant les entités).

Fournir de l'information agrégée et identifiée pose également question. En effet, cela signifie notifier à U_R que certaines entités bien identifiées sont impliquées dans des interactions mais sans préciser quelle entité est concernée par quelle interaction. Par exemple, informer U_R que certains des utilisateurs U_A , U_B et U_C ont déposé des nouvelles (*news*) et que certains ont envoyés des messages électroniques sans spécifier ce que chacun d'entre eux a fait.

Pour résumer, il apparaît que les informations identifiées et individuelles ou anonymes et agrégées sont les plus adéquates pour supporter la conscience de groupe. Par conséquent, nous scinderons la notion de conscience de groupe relative à l'espace de travail (*workspace awareness*) en deux concepts associés aux deux cas les plus utiles.

La conscience de groupe de l'espace de travail individuelle (*Workspace Individual Awareness* – WIA) sera associée à la communication à U_R d'informations identifiées et individuelles.

La conscience de groupe de l'espace de travail globale (*Workspace Global Awareness* – WGA) vise quant à elle à fournir une information globale sur l'espace collaboratif via de l'information agrégée et anonyme.

Habituellement, les collecticiels offrent des fonctionnalités pour supporter à la fois la WIA et la WGA. Dans certains cas, ces fonctionnalités sont couplées et permettent de passer d'une approche à l'autre. Par exemple, un graphique supportant la WGA (p.ex. icône indiquant que trois utilisateurs sont connectés) peut servir de point d'entrée pour accéder à des représentations orientées WIA (p.ex. les photos des trois personnes connectées).

Dans nos travaux, nous avons choisi de nous focaliser sur le concept de WGA, notamment parce qu'il a été moins étudié dans la littérature que celui de WIA (cf. Begole et al., 2003 ; Donath et al., 1999 ; Pinho et al., 2002 pour des exemples d'outils supportant WIA). En fait, supporter la WGA requiert de notifier à U_R de l'information globale sur les activités de l'espace de travail. La problématique des métriques de l'activité coopérative se pose donc tout naturellement. Cet aspect fera l'objet des sections suivantes.

3. Métriques d'activités coopératives

3.1 Travaux antérieurs

Evaluer le degré de coopération au sein d'un groupe est une tâche complexe, notamment à cause des difficultés associées à la conception de métriques significatives, à la collecte des données sous-jacentes à ces métriques et à l'interprétation des valeurs résultantes. Néanmoins certains chercheurs ont proposé de tels indicateurs.

Les méthodes les plus simples se basent sur le calcul du nombre d'occurrences de certains types d'interactions au sein du groupe. Nous pouvons en mentionner quelques exemples.

- Prinz et al. (2004) rapporte que le flux de courriels dans un projet peut être interprété comme un indicateur du niveau d'activité.
- Le système '*Participameter*' (Borges et Pino, 1999) est une autre approche basée sur le niveau relatif de participation de chaque membre dans un groupe de discussion.
- L'affichage du nombre d'utilisateurs en ligne, indisponibles et déconnectés a aussi été utilisé dans certaines applications pour supporter la conscience de groupe (Begole et al., 2003).
- Le logiciel *Community Toolbar* (Prinz et al., 2002) est une solution relativement avancée qui permet à U_R de choisir de quelles données relatives à la conscience de groupe il souhaite être tenu informé (p. ex. le nombre total d'utilisateurs connectés dans toutes les communautés dont il est membre, le nombre d'utilisateurs en train de visiter une communauté donnée ou le nombre de visiteurs d'une page web particulière).

D'autres chercheurs ont essayé d'élaborer des métriques reposant sur un traitement plus complexe des données brutes. Parmi ces travaux, ceux associés à la théorie des réseaux sociaux (*Social Network Analysis* – SNA) sont incontournables. Cette théorie se fonde sur la

similarité entre la théorie des graphes et la structure des relations au sein d'un groupe pour explorer la signification sociale de propriétés mathématiques du graphe des relations (p.ex. la notion de centralité). Certains concepts de la SNA ont par exemple été utilisés pour mesurer et visualiser le niveau d'activité d'étudiants dans une classe (Saltz et al., 2004). Barros et Verdejo (1999) ont adopté une autre approche. Ils ont utilisé un processus basé sur une chaîne d'inférences pour dériver des métriques de haut-niveau (p.ex. niveau de coopération) à partir de descriptions quantitatives mais aussi qualitatives des contributions des membres d'un groupe dans un contexte d'apprentissage collaboratif.

3.2 Proposition de métriques

3.2.1. Motivation et identification des métriques

Notre objectif consiste à étudier l'activité de coopération qui se réalise via un ensemble d'outils informatiques mis à disposition d'un groupe de personnes. Nous avons expliqué précédemment qu'il nous semblait opportun d'étudier la problématique d'un certain type de conscience de groupe, que nous avons appelé WGA.

Nous sommes donc confrontés au problème de définir **une métrique du niveau de coopération informatisée au sein d'un groupe**.

Cette métrique, que nous appellerons '**Indice composite global de coopération électronique**' (*Glocoopex* – *Global electronic Cooperation Composite Index*), a pour vocation de qualifier une situation coopérative utilisant des outils informatiques pour permettre de la comparer globalement avec une autre situation similaire. Cette comparaison se veut globale et il est entendu que, par définition, l'indice ne pourra pas refléter toute la complexité subtile des situations étudiées. Il faut le considérer comme un indice semblable aux indices boursiers qui indiquent une tendance mais ne reflètent pas le comportement individuel d'actions particulières. Pour des raisons de lisibilité, nous l'appellerons '**G**' dans la suite du texte²¹.

Rappelons que nous ne cherchons pas à ce stade à définir comment nous allons calculer la valeur de la métrique mais à définir ce que nous allons mesurer.

Nous considérerons fondamentalement que le niveau de coopération au sein d'un groupe se décompose en deux éléments. D'une part, il dépend du nombre d'interactions observées dans le groupe et d'autre part, il est influencé par la nature 'coopérative' de ces interactions.

Nous construirons donc notre indice *G* en combinant deux autres indices : un indice composite du niveau d'activité (*electronic Cooperation Activity Composite Index* – *Coopadex*) qui sera représentatif du nombre moyen d'interactions observées et un indice composite de l'intérêt des participants dans l'activité coopérative (*electronic Cooperation Interest Composite Index* – *Coopidex*) qui reflètera la nature coopérative moyenne des interactions. Par souci de lisibilité, nous utiliserons la notation '**Qt**' pour l'indice *Coopadex* et '**Qual**' pour l'indice *Coopidex*.

$$G = f(Q_t, Qual)$$

En conclusion, nous avons indentifié trois métriques (que nous avons appelées indices) qu'il nous faudra implémenter ensuite sous la forme de mesures. Nous chercherons donc à évaluer :

- le niveau d'activité au sein du groupe,
- l'intérêt des participants à créer des interactions de nature coopérative et
- le niveau de coopération au sein du groupe.

²¹ Les noms '*Glocoopex*', '*Coopadex*' et '*Coopidex*' sont utilisés *in extenso* dans l'article présenté à la conférence CRIWG 2006.

3.2.2. Comparabilité des métriques : le problème du temps

Le paramètre 'temps' joue évidemment un rôle décisif dans l'étude d'une situation de coopération. La comparaison de niveaux moyens d'activité coopérative n'a de sens que s'ils concernent des périodes de temps comparables. Cette question de la comparabilité n'est cependant pas triviale et mérite quelques développements. A première vue, on pourrait penser que l'égalité des durées absolues des périodes considérées (p.ex. périodes de trois mois) constitue le critère d'acceptation le plus naturel de la comparaison.

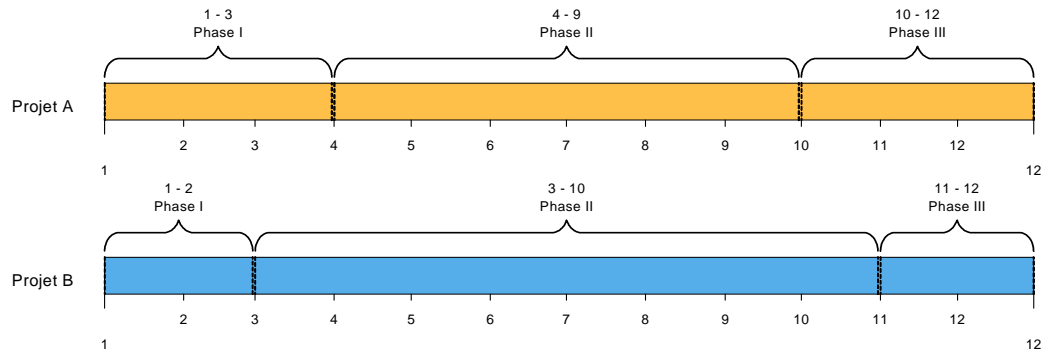


Fig. 74: Comparabilité de périodes de temps : exemple de projets informatiques

Après réflexion, on remarque cependant que, dans certaines circonstances, d'autres éléments déterminent de manière plus significative les bornes des périodes comparables. Prenons l'exemple de deux projets informatiques A et B composés chacun de trois phases : (I) analyse des besoins, (II) développement et (III) tests. La durée de chaque phase est indiquée sur la Fig. 74.

Dans ce cas de figure, est-il plus cohérent de comparer l'activité coopérative sur les deux premiers mois de chaque projet ou pendant les deux phases d'analyse des besoins ? Dans la première alternative, la durée absolue des périodes de comparaison est identique (c.-à.d. deux mois) mais elles représentent des fractions différentes de la phase d'analyse des besoins. Dans la seconde alternative, la comparaison porte sur l'entièreté de la phase d'analyse des besoins. Par contre, la durée absolue des périodes comparées est différente (trois mois *versus* deux mois).

En fait, nous pensons que, selon les cas, les deux approches peuvent être acceptables. A nos yeux, le critère essentiel à respecter est de s'interroger, de manière critique, sur les situations à comparer. S'il s'agit de situations dont la décomposition en phases distinctes est évidente, la comparaison de périodes fonctionnellement identiques est à privilégier. C'est notamment le cas si on étudie des projets. Si le but de la recherche est de comparer des situations de régime qui ne se présentent pas a priori comme une succession de sous-périodes, alors le choix de la durée absolue des périodes pour les délimiter est préférable. On peut citer comme exemple d'une telle situation l'étude d'un forum de discussion sur lequel des personnes communiquent chaque jour sans contraintes de planification particulières.

Pour répondre à cette dualité, nous introduisons le concept de '**durée sémantique**' qui englobe à la fois la notion habituelle de durée absolue et celle de durée fonctionnelle. La durée sémantique sera égale à la durée absolue dans le cas où des situations de régime sont comparées et elle sera fixée arbitrairement à la valeur unitaire pour des comparaisons de situations phasées.

La durée sémantique s'exprimera donc également en unités de temps.

Nous introduisons un formalisme spécifique pour identifier ce nouveau concept.

ΔT : durée sémantique de la période concernée

On note aisément la relation avec la notion habituelle de durée absolue d'une période :

$$\Delta T = (t_2 - t_1) : \text{durée absolue de la période concernée}$$

$\Delta^* T = \Delta T$ dans les situations où une comparaison des durées absolues est significative

$\Delta^* T = 1$ dans les situations où une comparaison des durées fonctionnelles est significative

3.2.3. Contraintes de base

Après avoir identifié des métriques et avoir discuté la problématique temporelle, il nous apparaît important de préciser la philosophie de base de notre démarche. Nous cherchons à **identifier des propriétés des interactions qui soient pertinentes pour caractériser le niveau de coopération présent dans les situations étudiées.**

Avant de décrire ces propriétés individuellement, il nous semble important de discuter les contraintes générales éventuelles qu'elles devraient satisfaire en vue de construire un modèle cohérent.

Ces propriétés devront satisfaire plusieurs contraintes fondamentales.

1. Tout d'abord, nous voulons limiter autant que possible les risques d'imprécisions ou de confusion dans les données d'entrée du modèle. Dès lors, il devra être possible de **fixer la valeur de toute propriété de manière indiscutable pour toute interaction**, ce que nous appellerons '**contrainte d'objectivité**'.
2. Ensuite, nous souhaitons pouvoir partitionner entièrement l'ensemble des interactions en sous-ensembles mutuellement exclusifs pour chaque axe d'analyse. Ceci se traduit en imposant que **pour toute interaction, une et une seule valeur soit toujours attribuée à chaque propriété**. Nous définissons ainsi une '**contrainte d'unicité**'.
3. De plus, comme notre but est de caractériser le niveau de coopération, nous ne considérerons comme acceptables que des **propriétés dont les valeurs permises sont associées à des niveaux ordonnables de coopération**. En d'autres termes, il devra toujours être possible de décider si la valeur A d'une propriété est associée à un niveau de coopération supérieur ou inférieur à celui de la valeur B. Nous nommerons cette contrainte : '**contrainte d'ordonnement**'.

Il est évident qu'un certain nombre de paramètres permettant de décrire les interactions se produisant dans une situation de coopération ne répondent pas à ces contraintes. Nous mentionnons ci-dessous pour chaque contrainte un exemple de propriété qui ne la satisfait pas.

- Une propriété qui renseignerait sur le sentiment (colère, satisfaction, indifférence...) associé à une interaction ne satisferait pas la *contrainte d'objectivité* car il est évident que l'affectation de la valeur de cette propriété résulterait d'un choix discutable par nature.
- Une propriété qui spécifierait le nom des documents attachés à un message électronique ne répondrait pas à la *contrainte d'unicité* car elle pourrait prendre plusieurs valeurs différentes pour une seule interaction.
- Une propriété qui identifierait le moment de la journée auquel l'interaction a lieu ne respecterait pas la *contrainte d'ordonnement* car rien ne permet d'affirmer, par exemple, qu'une interaction se produisant le matin est le signe d'une coopération plus ou moins intense qu'une autre se produisant l'après-midi.

Nous ne prétendons pas que de telles propriétés ne sont pas pertinentes ou utiles dans certains contextes. Nous pensons simplement qu'elles ne peuvent pas être employées pour définir une

métrique du niveau de coopération au sein d'un groupe qui soit à la fois simple, cohérente et portable au niveau opérationnel.

3.2.4. Indice Coopadex – Q_t

Nous avons conçu un premier indice : 'indice composite du niveau d'activité coopérative électronique' : ' Q_t '. Il tente de concentrer dans une seule valeur une réalité complexe présentant plusieurs dimensions.

L'indice Q_t a pour objectif d'être **représentatif du niveau d'utilisation moyen d'un ensemble d'outils informatiques mis à disposition d'un groupe pour le supporter dans la réalisation d'une activité coopérative**. Nous devons donc définir le concept de niveau d'utilisation de cet ensemble d'outils.

Pour ce faire, nous nous baserons sur les deux notions fondamentales sur lesquelles repose notre conceptualisation de la coopération : les *entités* et les *interactions*.

L'indice Q_t sera défini comme le nombre moyen d'interactions informatisées par membre d'un groupe donné pour une période déterminée.

Il se calcule très simplement en divisant le nombre total d'interactions informatisées par le nombre de membres du groupe et par la durée sémantique de la période étudiée. Formellement, cette définition se traduit comme suit :

- Soit Grp : un groupe de n personnes réalisant une activité coopérative
- Soit Grp_j : le j^{ème} membre du groupe Grp
- Soit I (Grp_j) : le nombre d'interactions impliquant le j^{ème} membre du groupe Grp
- Soit Δ^T : durée sémantique de la période sur lequel porte l'indice

$$Q_t = \frac{\sum_j (I(\text{Grp}_j))}{n \Delta^T}$$

L'indice Q_t a donc pour unités : [interaction . personne⁻¹ . unité de temps⁻¹]

On remarque que le dénominateur contient le facteur n , c.-à-d. le nombre total de membres dans le groupe Grp et non le nombre de membres qui sont effectivement impliqués dans des interactions informatisées. Il s'agit en effet de proposer un indice du niveau d'activité au sein de tout le groupe.

Il est aussi important de noter que l'on additionne le nombre d'interactions pour chaque membre du groupe. En d'autres termes, si une interaction est initiée par le membre Grp_j à destination des membres Grp_k et Grp_m, cette interaction sera comptabilisée trois fois dans l'indice. Ce mode de calcul se justifie par le fait que l'indice est destiné à refléter un niveau moyen d'interactions par personne. Une interaction qui concerne trois personnes sera plus valorisée qu'une autre qui en implique seulement deux car elle génère une plus grande activité au sein du groupe. Comme nous l'expliquerons plus tard dans la description des autres indices, cette approche permettra également une catégorisation aisée des interactions.

3.2.5. Indices Coopindex – Qual et N-Qual

L'indice *Qt* reflète le taux d'utilisation moyen des outils de l'environnement de coopération. Il ne transcrit pas, par contre, le degré d'implication, d'intérêt ou de motivation des membres du groupe envers ces outils. Nous avons donc réfléchi à la manière de combler cette lacune.

Nous avons défini un nouvel indice : l'indice composite de l'intérêt dans l'activité coopérative électronique : '*Qual*'.

L'idée de base consiste à calculer une somme pondérée de divers types d'interactions qui tiennent compte du niveau d'intérêt envers la coopération auquel chaque type peut être associé.

La principale difficulté rencontrée pour concevoir cet indice réside dans l'évaluation de la nature coopérative des interactions électroniques. Nous rappelons à cet égard que les propriétés des interactions qui sont éligibles pour composer l'indice doivent satisfaire les contraintes d'objectivité, d'unicité et d'ordonnancement (cf. p. 155).

Au terme de notre réflexion, seules deux propriétés ont été retenues.

Tout d'abord, la *nature facultative ou imposée* d'une interaction électronique nous semble être une première propriété pertinente.

- Elle satisfait la contrainte d'objectivité car il est toujours possible de déterminer si le fait d'initier une interaction ou d'en être la cible est imposé ou optionnel pour une entité. Par exemple, un message électronique de demande d'information est une interaction optionnelle alors que le dépôt obligatoire d'un document dans un système partagé avant une date fixée est une interaction imposée.
- Elle satisfait la contrainte d'unicité car une interaction électronique ne peut être simultanément imposée et facultative. Rappelons à cet égard l'importance de notre remarque antérieure sur la manière de comptabiliser les interactions dans l'indice *Qt* (cf. p. 156). Une interaction concernant plusieurs entités doit être considérée comme autant d'interactions différentes. Par ce mécanisme, l'expédition d'un message électronique peut être facultative pour l'entité initiatrice alors que sa réception est imposée à l'entité ciblée.
- Elle satisfait la contrainte d'ordonnancement. En effet, on peut légitimement considérer que la présence d'interactions facultatives reflète un plus grand degré de motivation envers la coopération de la part des entités que celle d'interactions imposées puisque dans le premier cas il s'agit d'un choix délibéré.

En second lieu, la *nature passive ou active* de l'interaction électronique apparaît comme une autre propriété significative. On définira une interaction active comme une interaction dont l'occurrence nécessite une action explicite de l'entité concernée (p.ex. envoyer un courriel). A contrario, une interaction passive ne nécessite pas d'action explicite (p.ex. recevoir un courriel).

- Cette propriété satisfait la contrainte d'objectivité car il est toujours possible de déterminer si une interaction a nécessité une action explicite de l'entité concernée ou non.
- Elle satisfait la contrainte d'unicité car une interaction électronique ne peut être simultanément active et passive.
- Elle satisfait la contrainte d'ordonnancement. En effet, les utilisations actives sont un signe de plus grande implication dans la coopération électronique que les utilisations passives car elles ont nécessité un effort de la part de l'entité initiatrice.

Afin de répondre à la contrainte de distinguer les interactions informatisées optionnelles de celles qui sont imposées aux entités, nous introduisons les coefficients α_o (optionnel) et α_i (imposé). Ces deux coefficients peuvent prendre une valeur comprise entre 0 et 1, selon l'importance que l'on souhaite accorder à cette catégorisation des interactions mais, conformément à notre discussion préalable, nous imposons : $\alpha_o > \alpha_i$.

Afin de répondre à la contrainte de distinguer les interactions informatisées actives et passives, nous introduisons les coefficients β_a (interaction active) et β_p (interaction passive). Ces deux coefficients peuvent prendre une valeur comprise entre 0 et 1, selon l'importance que l'on souhaite accorder à cette catégorisation des interactions mais, conformément à notre discussion préalable, nous imposons : $\beta_a > \beta_p$.

Pour calculer l'indice *Qual*, nous devons tout d'abord catégoriser les interactions informatisées selon les deux axes de différenciation : « *interaction optionnelle – interaction imposée* » et « *interaction active – interaction passive* ». Nous obtenons donc quatre nombres :

- $I_{a,o}$: le nombre d'interactions actives optionnelles
- $I_{p,o}$: le nombre d'interactions passives optionnelles
- $I_{a,i}$: le nombre d'interactions actives imposées
- $I_{p,i}$: le nombre d'interactions passives imposées

Nous calculons ensuite l'indice *Qual* par l'expression :

$$Qual = \frac{\alpha_o \beta_a I_{a,o} + \alpha_o \beta_p I_{p,o} + \alpha_i \beta_a I_{a,i} + \alpha_i \beta_p I_{p,i}}{I_{a,o} + I_{p,o} + I_{a,i} + I_{p,i}}$$

Il est utile de noter que $(I_{a,o} + I_{p,o} + I_{a,i} + I_{p,i}) = \sum_j (I(Grp_j))$ car il s'agit naturellement des mêmes interactions.

L'indice *Qual* a pour unités [interaction . interaction⁻¹]. Il s'agit donc d'une grandeur sans dimensions. Il peut être interprété comme une métrique de la motivation moyenne des membres du groupe à générer des interactions en vue de coopérer électroniquement.

Cependant, il reste un problème que nous n'avons pas encore traité. L'indice *Qual* voit sa valeur varier entre la borne inférieure ($\alpha_i \beta_p$) dans le cas où toutes les interactions informatisées sont imposées et passives et la borne supérieure ($\alpha_o \beta_a$) dans le cas où toutes les interactions informatisées sont optionnelles et actives. La valeur de l'indice va donc être assez difficile à interpréter vu qu'elle nécessite de connaître les valeurs des paramètres sous-jacents : α_i , α_o , β_a et β_p .

Nous souhaitons normaliser cette valeur en la projetant sur l'intervalle [0,1]. Cette opération a pour but essentiel de construire un indice normalisé plus facile à interpréter.

Nous définissons donc un nouvel indice *N-Qual* que nous calculons avec l'expression :

$$N-Qual = \frac{(\alpha_o \beta_a - \alpha_i \beta_p)}{(\alpha_o \beta_a - \alpha_i \beta_p)}$$

Si on développe la définition de l'indice *Qual*, cette expression se réécrit comme suit.

$$N-Qual = \frac{[(\alpha_o \beta_a - \alpha_i \beta_p) I_{a,o} + \beta_p (\alpha_o - \alpha_i) I_{p,o} + \alpha_i (\beta_a - \beta_p) I_{a,i}]}{(I_{a,o} + I_{p,o} + I_{a,i} + I_{p,i}) (\alpha_o \beta_a - \alpha_i \beta_p)}$$

L'indice *N-Qual* varie entre la valeur nulle qui correspond au cas où toutes les interactions informatisées sont passives et obligatoires et la valeur unitaire qui correspond au cas où toutes les interactions informatisées sont actives et optionnelles. Plus la valeur de l'indice *N-Qual* est élevée, plus le groupe génère activement des interactions électroniques à la propre initiative de ses membres.

L'indice *N-Qual* est lui aussi une grandeur sans dimensions.

A ce niveau, il est important de rappeler que les indices relatifs à deux situations données ne seront comparables que s'ils se réfèrent à des périodes de temps comparables (cf. p. 154) et si les coefficients α_o , α_i et β_a , β_p utilisés pour les calculer sont identiques. A ces contraintes, s'ajoutent aussi l'exigence que les situations soient par nature comparables.

3.2.6. Indice Glocoopex – G

Nous disposons à présent de l'indice *Qt* qui nous renseigne sur le nombre moyen d'interactions informatisées par personne du groupe et par période de temps. D'autre part, l'indice *N-Qual* nous fournit un niveau moyen de motivation des membres du groupe par rapport à la coopération électronique.

Si on multiplie ces deux indices, on obtient un indicateur global de l'intensité des interactions informatisées qui tient aussi compte de l'intérêt des membres du groupe pour la coopération. Nous appellerons cet indice : *G (Global electronic Cooperation Composite Index)*.

$$G = Qt \cdot N-Qual$$

Si on développe cette expression sous forme complète, on obtient :

$$G = \frac{[(\alpha_o \beta_a - \alpha_i \beta_p) I_{a,o} + \beta_p (\alpha_o - \alpha_i) I_{p,o} + \alpha_i (\beta_a - \beta_p) I_{a,i}]}{n \Delta T (\alpha_o \beta_a - \alpha_i \beta_p)}$$

L'indice *G* a pour unités [interaction . personne⁻¹ . unité de temps⁻¹].

Pour le rendre plus précis et plus informatif, nous introduisons le concept d'équivalents-interactions de haute qualité coopérative (*Equivalent High Quality Interactions – EHQI*). Il traduit l'idée selon laquelle les interactions peuvent être comparées à des interactions abstraites qui sont théoriquement considérées comme les meilleures en termes de coopération.

Plus un ensemble d'interactions est proche du cas théorique idéal (dans notre modèle : les interactions actives optionnelles), plus il obtiendra un score élevé exprimé en EHQI. L'introduction de cette nouvelle unité est en fait un moyen de rendre visible la contribution de l'indice *N-Qual* dans la valeur de *G*.

Les unités de l'indice *G* deviennent alors : [EHQI . personne⁻¹ . unité de temps⁻¹].

Cette nouvelle unité EHQI attire aussi l'attention sur le fait que la valeur de l'indice *G* ne doit pas être comprise comme un nombre moyen d'interactions réelles par membre du groupe et par unité de temps mais un nombre moyen d'EHQI par membre du groupe et par unité de temps.

Le comportement de l'indice *G* peut aussi être étudié en calculant sa sensibilité relative aux variables : $I_{p,i}$, $I_{a,o}$, $I_{p,o}$ et $I_{a,i}$.

Tout d'abord, on peut voir que la variation de l'indice *G* par rapport au nombre d'interactions actives optionnelles $I_{a,o}$ ne dépend pas de la valeur choisie pour les coefficients α_o , α_i et β_a , β_p . En fait, comme nous l'expliciterons de manière détaillée dans la suite, le terme $I_{a,o}$ est celui qui influence le plus la valeur finale de l'indice *G*. Cette observation est conforme à la représentativité que nous souhaitons attribuer à cet indice.

Par ailleurs, l'indice *G* n'est pas sensible à la variable $I_{p,i}$. Ceci est la conséquence de la normalisation de l'indice *Qual* pour donner naissance à l'indice *N-Qual*. Ce comportement ne nous gêne pas car il revient à considérer que dans un contexte coopératif le fait que les membres du groupe soient impliqués des interactions passives en étant obligés de le faire ne peut être considéré comme un indicateur d'un niveau significatif d'activité coopérative. Par exemple, le fait que des courriels de notification des modifications d'un site web partagé soient reçus par les participants ne signifie pas que ceux-ci se sentent motivés à coopérer.

La dérivée partielle de *G* par rapport à la variable $I_{p,o}$ nous renseigne sur sa sensibilité à celle-ci.

$$\frac{\partial G}{\partial I_{p,o}} = \frac{\beta_p (\alpha_o - \alpha_i)}{n \Delta^T (\alpha_o \beta_a - \alpha_i \beta_p)}$$

Il nous semble légitime de nous demander si la sensibilité de l'indice *G* à une variation de $I_{p,o}$ est plus grande ou plus réduite par rapport à celle qu'il présente vis-à-vis de $I_{a,o}$. Pour répondre à cette question, nous devons évaluer si le facteur qui différencie les deux dérivées partielles est supérieur ou inférieur à la valeur '1'.

Essayons de démontrer que cette valeur est toujours inférieure à '1'.

$$\frac{\beta_p (\alpha_o - \alpha_i)}{(\alpha_o \beta_a - \alpha_i \beta_p)} < 1$$

Sur base des contraintes $\alpha_o > \alpha_i$ et $\beta_a > \beta_p$, nous déduisons que le facteur $(\alpha_o \beta_a - \alpha_i \beta_p)$ est toujours un nombre positif.

$$\Leftrightarrow \beta_p (\alpha_o - \alpha_i) < (\alpha_o \beta_a - \alpha_i \beta_p)$$

$$\Leftrightarrow (\alpha_o \beta_p - \alpha_i \beta_p) < (\alpha_o \beta_a - \alpha_i \beta_p)$$

$$\Leftrightarrow \alpha_o \beta_p < \alpha_o \beta_a$$

Nous savons également que $\alpha_o > 0$.

$$\Leftrightarrow \beta_p < \beta_a \quad \text{ce qui est une contrainte connue sur les coefficients } \beta_a, \beta_p$$

En conclusion, nous pouvons déduire que ce facteur est toujours inférieur à la valeur '1'. Dès lors, nous savons que l'indice G sera toujours moins sensible à une variation du nombre d'interactions passives optionnelles qu'à une variation du nombre d'interactions actives optionnelles. Cette conclusion est en phase avec la représentativité souhaitée de cet indice.

La dérivée partielle de G par rapport à la variable $I_{a,i}$ nous renseigne sur sa sensibilité à celle-ci.

$$\frac{\partial G}{\partial I_{a,i}} = \frac{\alpha_i (\beta_a - \beta_p)}{n \Delta^T (\alpha_o \beta_a - \alpha_i \beta_p)}$$

Il nous semble légitime de nous demander si la sensibilité de l'indice G à une variation de $I_{a,i}$ est plus grande ou plus réduite par rapport à celle qu'il présente vis-à-vis de $I_{a,o}$. Pour répondre à cette question, nous devons évaluer si le facteur qui différencie les deux dérivées partielles est supérieur ou inférieur à la valeur '1'.

Essayons de démontrer que cette valeur est toujours inférieure à '1'.

$$\frac{\alpha_i (\beta_a - \beta_p)}{(\alpha_o \beta_a - \alpha_i \beta_p)} < 1$$

Sur base des contraintes $\alpha_o > \alpha_i$ et $\beta_a > \beta_p$, nous déduisons que le facteur $(\alpha_o \beta_a - \alpha_i \beta_p)$ est toujours un nombre positif.

$$\Leftrightarrow \alpha_i (\beta_a - \beta_p) < (\alpha_o \beta_a - \alpha_i \beta_p)$$

$$\Leftrightarrow (\alpha_i \beta_a - \alpha_i \beta_p) < (\alpha_o \beta_a - \alpha_i \beta_p)$$

$$\Leftrightarrow \alpha_i \beta_a < \alpha_o \beta_a$$

Nous savons également que $\beta_a > 0$.

$$\Leftrightarrow \alpha_i < \alpha_o \quad \text{ce qui est une contrainte connue sur les coefficients } \alpha_i, \alpha_o$$

En conclusion, nous pouvons déduire que ce facteur toujours est inférieur à la valeur '1'. Dès lors, nous savons que l'indice G sera toujours moins sensible à une variation du nombre d'interactions actives imposées qu'à une variation du nombre d'interactions actives optionnelles. Cette conclusion est en phase avec la représentativité souhaitée de cet indice.

Un autre question intéressante à étudier concerne la sensibilité relative de l'indice G à aux variations de $I_{p,o}$ et de $I_{a,i}$. Nous devons donc comparer les deux dérivées partielles associées.

$$\frac{\partial G}{\partial I_{p,o}} = \frac{\beta_p (\alpha_o - \alpha_i)}{n \Delta^T (\alpha_o \beta_a - \alpha_i \beta_p)}$$

et

$$\frac{\partial G}{\partial I_{a,i}} = \frac{\alpha_i (\beta_a - \beta_p)}{n \Delta^T (\alpha_o \beta_a - \alpha_i \beta_p)}$$

Si on pose comme hypothèse que les interactions passives optionnelles représentent une motivation supérieure de la part des membres du groupe que les interactions actives imposées, quelles sont les conséquences sur les valeurs des coefficients ?

$$\frac{\beta_p (\alpha_o - \alpha_i)}{n \Delta^T (\alpha_o \beta_a - \alpha_i \beta_p)} > \frac{\alpha_i (\beta_a - \beta_p)}{n \Delta^T (\alpha_o \beta_a - \alpha_i \beta_p)}$$

Comme les facteurs n , Δ^T et $(\alpha_o \beta_a - \alpha_i \beta_p)$ sont toujours positifs, nous pouvons écrire :

$$\beta_p (\alpha_o - \alpha_i) > \alpha_i (\beta_a - \beta_p)$$

$$\Leftrightarrow (\alpha_o \beta_p - \alpha_i \beta_p) > (\alpha_i \beta_a - \alpha_i \beta_p)$$

$$\Leftrightarrow \alpha_o \beta_p > \alpha_i \beta_a$$

Nous imposerons donc cette nouvelle contrainte sur les coefficients. Elle conduira à considérer que la présence d'interactions informatisées passives optionnelles au sein d'un groupe est un meilleur indicateur de la motivation de ses membres à collaborer via des outils du collecticiel que la présence d'interactions informatisées actives imposées.

Pour prendre un exemple, si les membres d'un groupe choisissent d'être appelables par le système de *chat* commun ou s'ils optent pour recevoir les informations de services par la

messagerie électronique utilisée par le groupe, nous considérerons qu'ils sont plus motivés pour collaborer électroniquement que s'ils déposent sur le site web du groupe des documents qu'ils sont obligés d'y placer.

Néanmoins, nous sommes conscients qu'il s'agit d'un choix de notre part. Nous avons choisi de privilégier la motivation individuelle. En effet, la décision de participer, même passivement, à une interaction sans y être obligé relève de la motivation personnelle qui est selon nous un indicateur significatif de la volonté de participer à la coopération.

3.2.7. Remarque finale sur la portée du modèle de métrique

Dès le départ, nous avons limité la portée du modèle de métrique à des situations de coopération reposant sur des outils informatiques. Cette réduction de son champ d'action fut principalement motivée par deux éléments. D'une part, notre travail a pour objet la visualisation de contextes de travail coopératif assisté par l'informatique. D'autre part, nous nous focalisons uniquement sur les interactions informatisées.

Cependant, nous pensons que le modèle peut s'appliquer à toute situation de coopération. En effet les propriétés sur lesquelles se basent les indices ne sont pas spécifiques aux interactions informatisées. Elles s'appliquent avec autant de pertinence à des réunions en face-à-face qu'à des échanges de courriels. Une généralisation de notre approche pour étudier de manière plus globale des situations de coopération partiellement, voire nullement informatisée nous apparaît comme une piste de recherche prometteuse. Elle implique néanmoins un travail sociologique ou ethnologique qui dépasse le cadre et les objectifs de notre travail.

3.2.8. Illustration des indices avec des données simulées

Afin d'illustrer l'intérêt des indices définis précédemment, nous proposons un exemple basé sur des valeurs acceptables des différents coefficients : $\alpha_o = 0,8$; $\alpha_i = 0,2$; $\beta_a = 0,6$; $\beta_p = 0,3$.

La Table 12 indique la valeur des indices pour une situation de coopération simulée. Nous considérerons que les données correspondent à une période de temps d'une unité ($\Rightarrow \Delta T = 1$) et que le groupe compte 10 membres ($\Rightarrow n=10$).

La table montre bien que le nombre d'interactions doit être pondéré par leur qualité en termes de coopération. Alors qu'en moyenne les membres du groupe sont impliqués dans 12,20 interactions, la prise en compte de la qualité de celles-ci montre qu'ils ne sont concernés que par 3,20 EHQI.

On note également que l'indice G est plus informatif en l'exprimant avec les unités [EHQI . personne⁻¹ . unité de temps⁻¹] qu'avec les unités [interactions . personne⁻¹ . unité de temps⁻¹].

Table 12: Simulation d'une situation de coopération

Répartition des interactions			
Interactions	Imposées	Optionnelles	Total
Actives	21	17	38
Passives	56	28	84
Total	77	45	122
Valeur des indices			
Q_t	$Qual$	$N-Qual$	G
12,20	0,17	0,262	3,20

4. Visualisation des indices d'activité

Nous avons précédemment défini des indices de l'activité coopérative. Comme notre but est d'étudier la manière de visualiser des informations relatives à une plate-forme électronique de coopération, il était logique de commencer par représenter la valeur de ces indices.

4.1 Approche technologique

Nous avons appliqué l'approche théorique décrite précédemment au cas d'une plate-forme de coopération accessible via web. Cette plate-forme disposait des fonctionnalités suivantes : partage de documents, agenda des activités, liste de diffusion par courriels, publication de nouvelles (*news*).

Notre application prototype de visualisation des indices d'activité repose sur une architecture informatique à trois composants (cf. Fig. 75).

- **La plate-forme électronique collaborative** basée sur les technologies habituelles Apache, PHP, MySQL. Sur la base de données MySQL (A) ont été ajoutés des *triggers* qui sont déclenchés lorsque des interactions sont initiées. Ils alimentent une seconde base de données MySQL (B). La liste de diffusion fait appel à des fichiers 'texte' qui sont lus par un module dédié écrit en langage PHP. Ce dernier insère la base de données B les interactions correspondantes.
- **Un module de calcul** des indices d'activité écrit en langage Java. Ce module interroge la seconde base de données (B) dans laquelle est implémenté notre modèle de la coopération basé sur des entités et des interactions. Des tables sont également prévues pour stocker les valeurs des indices d'activité qui sont calculés.
- **Un module de visualisation** des indices d'activité écrit en langage Java. Il interroge la base de données B pour lire les valeurs à afficher.

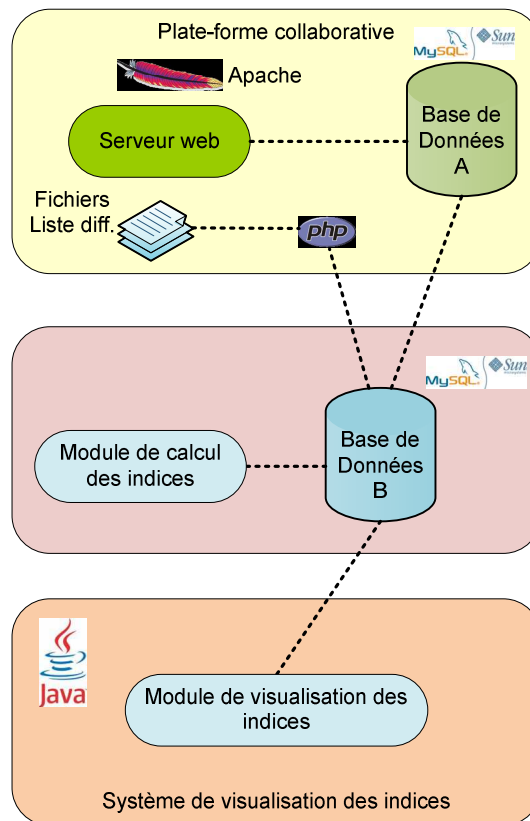


Fig. 75: Architecture informatique du système de visualisation des indices d'activité

4.2 Qualification des interactions

Outre le volet technologique, la mise en œuvre de notre approche théorique de calcul d'indices nécessite de qualifier les interactions selon les deux axes de classification retenus.

En pratique, les règles détaillées dans la Table 13 peuvent être utilisées pour catégoriser les interactions de façon automatique. Elles sont relativement simples mais présentent l'avantage d'être automatisables et de fournir une image globale raisonnablement correcte.

A la lecture de la Table 13, on pourrait croire que quasiment toutes les interactions sont actives et facultatives. En pratique, ce n'est pas tout à fait le cas car les courriels représentent souvent une part importante des interactions.

Table 13: Qualification des interactions pour le calcul des indices d'activité

Type d'interaction	Axe 1		Axe 2	
	Active	Passive	Facultative	Imposée
Envoi de courriel par utilisateur	X		X	
Réception de courriel envoyé par utilisateur		X		X
Envoi de courriel généré automatiquement		X		X
Réception de courriel généré automatiquement		X		X
Dépôt de document (<i>upload</i>)	X		X	
Téléchargement de document (<i>download</i>)	X		X	
Visite page d'accueil du site avec résumé de l'agenda et des nouvelles	X		X	
Visite de la page complète des nouvelles (<i>news</i>)	X		X	
Publication d'une nouvelle (<i>add news</i>)	X		X	
Visite de la page complète de l'agenda	X		X	
Ajout d'une activité dans l'agenda	X		X	

Par ailleurs, on remarque que les règles spécifiques utilisées dans cet exemple ne couvrent que la moitié des cas possibles. Cette observation met en évidence le fait qu'un certain degré d'incertitude existe sur l'affectation des interactions. D'une manière générale, si des informations permettant de mieux catégoriser les interactions sont disponibles, les autres cas de figure seront naturellement utilisés. Par exemple, si les interactions sont issues d'un système de *workflow* qui précise explicitement la nature facultative ou obligatoire d'interactions telles que le dépôt d'un document, la catégorie 'active – imposée' pourrait être utilisée en sus de celle : 'active – facultative'.

D'autres développements complémentaires sont naturellement possibles en vue de classer plus finement les interactions. Malheureusement, pour mettre en place une telle granularité en pratique, il faudrait disposer d'informations détaillées sur le contenu des documents et sur leur rôle opérationnel. Bien que très intéressante en vue d'une étude sociologique ou managériale

de la coopération, cette approche aurait nécessité des ressources importantes, notamment pour la mise à jour de ces données.

En ce qui nous concerne, nous avons choisi de nous concentrer sur la thématique de la visualisation plutôt que sur celle de l'étude sociologique de la coopération électronique. Par conséquent, la classification présentée en Table 13 était suffisante pour nos travaux.

4.3 Evolution de l'activité coopérative

Le but de cette représentation graphique est de visualiser l'évolution des indices de la coopération au cours du temps. Il s'agit donc fondamentalement d'une visualisation d'historique. Dans ce contexte, l'originalité de cette première proposition concerne la définition des indices eux-mêmes. Nous avons choisi de ne pas innover en matière de visualisation à ce niveau. Cette décision était motivée par le fait que nous ne voulions pas exiger de l'utilisateur un effort trop important qu'aurait imposé une technique de visualisation inconnue pour représenter des données elles aussi conceptuellement nouvelles. Dès lors, nous avons adopté la technique classique de représentation par des courbes sur un plan à deux dimensions (cf. Fig. 76). L'axe horizontal est associé à la variable temporelle.

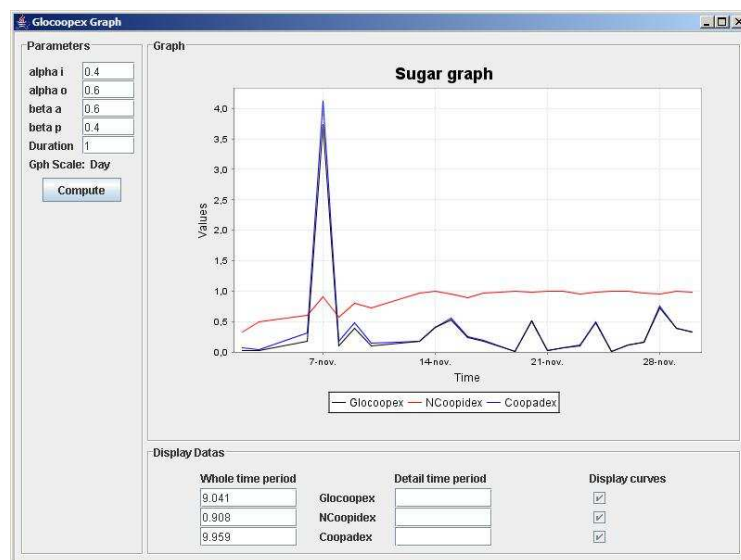


Fig. 76: Visualisation des indices d'activité coopérative (exemple 1)

Un prototype a été développé en langage Java pour matérialiser les concepts théoriques décrits dans les sections précédentes. Il permet de fixer les différents paramètres (α_i , α_o , β_a , β_p , unité de temps de référence, période considérée) nécessaires au calcul des indices de coopération (cf. Fig. 77).



a. Sélection de la période considérée



b. Sélection de la granularité

Fig. 77: Sélection des paramètres de calcul des indices d'activité coopérative

La fenêtre principale visualise l'évolution de la valeur des différents indices au cours de la période de temps sélectionnée. Les valeurs des trois indices G , $N-Qual$ et Qt sont visualisées par des courbes de couleurs différentes.

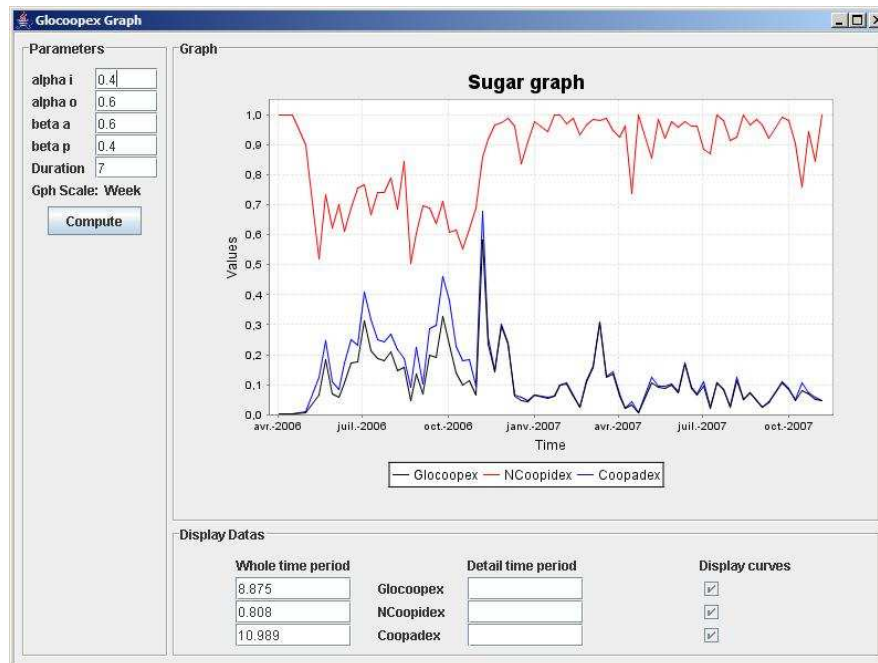


Fig. 78: Visualisation des indices d'activité coopérative (exemple 2)

Pour rappel, l'indice Qt est défini comme le nombre moyen d'interactions informatisées par membre d'un groupe donné pour une période déterminée. L'indice $N-Qual$ est un indice normalisé qui rend compte du caractère coopératif des interactions. Enfin, l'indice G est le produit des deux premiers indices et constitue un indicateur global de l'intensité des interactions des interactions informatisées qui tient aussi compte de l'intérêt des membres du groupe pour la coopération.

A titre illustratif, nous pouvons observer dans la Fig. 78 qu'au cours de la période représentée, l'indice $N-Qual$ s'est amélioré. Cela se traduit graphiquement par le fait que les courbes représentant les indices G et Qt sont presque superposées dans la partie droite de la figure.

4.4 Niveau d'activité courant

Dans la section précédente, nous expliquons comment nous visualisons l'évolution des indices d'activité au cours du temps. Ce module permet d'étudier la dynamique de l'activité.

Un autre élément peut également s'avérer utile pour les utilisateurs de la plate-forme : la visualisation du niveau d'activité courant²². Ce niveau est matérialisé dans nos travaux par les indices *Qt*, *N-Qual* et *G*. Pour rappel, ces valeurs sont disponibles dans la base de données B (cf. Fig. 75). Le système de visualisation du niveau d'activité courant interrogera donc cette base B.

Pour concevoir ce système, nous nous sommes basés sur l'idée de conscience périphérique (*peripheral awareness*) introduite par Heath et Luff (1991). Elle identifie six principes de conception qui ont été repris par Cadiz et al. (2002) dans leur système *Sideshow* :

1. **Présence permanente**
Inclure les visualisations dans des applications standard pour réduire le besoin d'utiliser des applications spécifiques pour obtenir de l'information associée à la conscience de groupe.
2. **Minimisation du mouvement**
Limiter dans les visualisations l'utilisation d'animations ou de rafraichissements à haute fréquence des données.
3. **Personnalisation**
Permettre aux utilisateurs de paramétrer ce qu'ils voient ainsi que d'autres éléments comme la fréquence de rafraichissement.
4. **Exensibilité**
Permettre la connexion avec différentes applications ou différentes sources de données.
5. **Affichage rapide des détails**
Il n'est pas possible d'afficher tous les détails des données dans une application supportant la conscience périphérique. Par contre, il doit être facile d'afficher les détails sur ces données puis de revenir à la tâche principale courante.
6. **Croissance**
Le système doit être capable de supporter une croissance des données à afficher.

Dans notre cas, tous les principes ne s'appliquent pas avec la même pertinence. Rappelons que nous voulons visualiser la valeur courante de trois indices *Qt*, *N-Qual* et *G*.

Dans notre contexte, la présence permanente et la minimisation du mouvement sont prépondérantes. Nous n'offrirons pas de véritable fonctionnalité de personnalisation car elle est (en partie) disponible dans l'application de visualisation de l'historique. Pour garantir l'extensibilité, il suffit d'ajouter des modules d'alimentation de la base de données B à partir d'autres applications. L'affichage rapide des détails n'est pas utile dans notre cadre car nous ne visons qu'à afficher des données agrégées. Le détail est disponible sur la plate-forme de coopération elle-même. Enfin, la problématique de la croissance des données ne se pose pas dans les termes habituels puisque les indices sont le résultat de calculs sur un volume potentiellement grand de données. Au niveau de la visualisation, nous ne cherchons qu'à afficher trois valeurs, indépendamment du volume de données brutes à partir desquelles elles sont dérivées. Par contre, il est tout à fait possible que l'utilisateur souhaite rester informé du niveau d'activité de plusieurs plates-formes. Nous devons donc être capables d'afficher simultanément plusieurs représentations graphiques indépendantes.

Pour satisfaire les différentes contraintes citées précédemment, nous avons choisi d'utiliser la technologie *Windows Active Desktop*. Elle permet l'affichage de graphiques directement dans

²² En réalité, les indices courants sont calculés sur base d'un intervalle de temps correspondant à un passé récent.

l'interface du système d'exploitation²³. Les ajouts apparaissent donc comme des parties intégrantes du 'bureau Windows' au même titre que des icônes de démarrage d'applications. Ils peuvent donc être (re)positionnés selon le bon vouloir de l'utilisateur. Ces graphiques peuvent être dynamiquement rafraîchis sans intervention de l'utilisateur en pointant vers une source placée sur un serveur dédié à la mise à jour du graphique. Enfin, les graphiques ne disposent pas d'une réelle interface de type GUI et ne nécessite donc aucune intervention de la part de l'utilisateur.

Nous avons choisi d'adopter la métaphore d'un drapeau pour représenter les différents indices. Cette décision fut prise parce que dans beaucoup de situations les drapeaux sont associés à la communication d'une information. Par exemple, ils sont utilisés pour communiquer entre navires dans la marine ou pour informer les pilotes d'événements particuliers pendant une compétition automobile. De plus, nous souhaitions utiliser une métaphore facile à comprendre apte à supporter WGA dans un mode de conscience périphérique. Cette technique de visualisation fut nommée '*Ambient Flag*'.

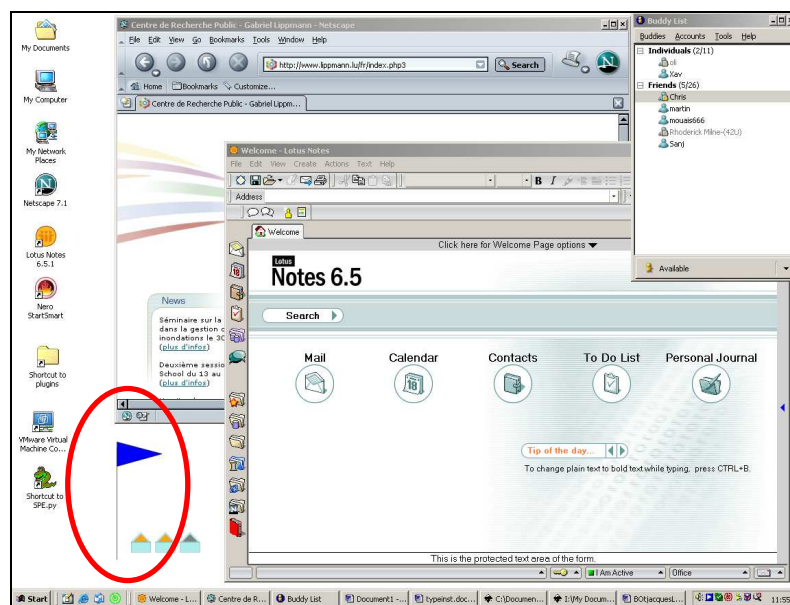


Fig. 79: Application de visualisation de l'activité courante (*Ambient Flag*)

Grâce à la technologie *Windows Active Desktop*, le(s) drapeau(x) sont placés directement sur le bureau de l'ordinateur (cf. Fig. 79). L'utilisateur ne voit s'exécuter aucune application, ce qui est cohérent avec la philosophie de la conscience périphérique dans laquelle l'utilisateur peut rapidement obtenir de l'information annexe (c.à-d. ici le niveau d'activité sur une plateforme collaborative) sans abandonner trop longtemps sa tâche courante.

Il nous faut à présent expliquer comment nous avons associé certains paramètres du drapeau aux trois indices d'activité.

Des recherches antérieures ont mis en évidence les concepts de dimensions visuelles intégrales et séparables (cf. Ware, 2004, p. 176-183). Des dimensions visuelles intégrales sont perçues de manière holistique. Par contre, en présence de dimensions visuelles séparables, l'utilisateur a tendance à faire des jugements séparés à propos de chaque dimension. Ware fournit une liste de paires de dimensions visuelles ordonnées depuis les plus intégrales jusqu'au plus séparables (cf. Fig. 80).

²³ Cette technologie n'est disponible que si le système d'exploitation *Microsoft Windows* est installé sur l'ordinateur.

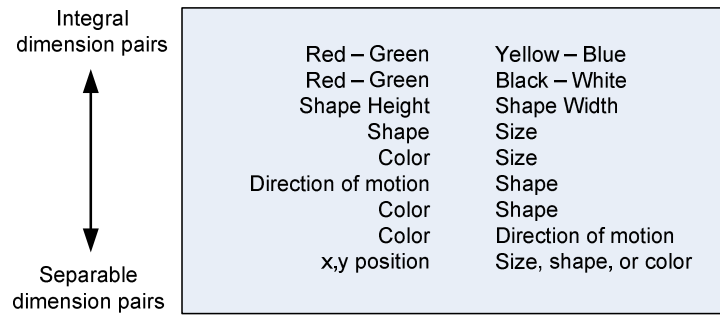


Fig. 80: Classement des paires de dimensions visuelles (source : [212], p. 180).

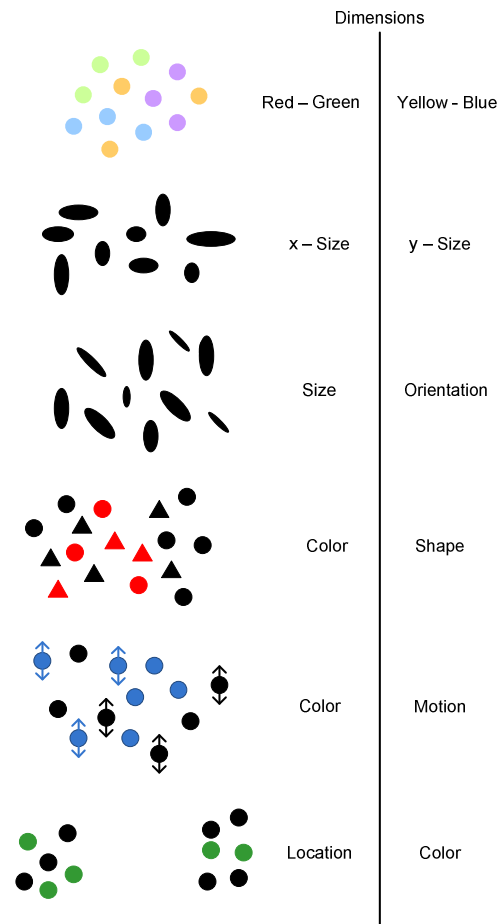


Fig. 81: Exemple de paires de variables visuelles classées selon l'axe 'intégral – séparable' (source : [212], p. 181).

Il illustre la liste théorique par une série d'exemples (cf. Fig. 81). On remarque ainsi que dans la paire (*Red – Green, Yellow – Blue*), il est très difficile de percevoir que deux informations différentes sont codées alors que cela est beaucoup plus évident dans le cas de la paire (*Color, Shape*).

Sur base de ces résultats, il apparaît que la hauteur et la largeur d'une forme composent une paire intégrale de dimensions. En d'autres termes, si nous voulons que les indices *Qt* et *N-Qual* soient interprétés globalement, il est justifié de les associer à ces deux dimensions. Cette approche est particulièrement cohérente si l'intention est que l'information soit comprise de

manière holistique et rapide. Par conséquent, nous avons associé la longueur du drapeau (axe horizontal) à l'indice Qt et la hauteur²⁴ du drapeau (axe vertical) à l'indice $N-Qual$.

Le niveau global d'activité (G) est défini comme le produit des indices Qt et $N-Qual$. Suite à notre mise en correspondance de ces indices avec la largeur et la hauteur du drapeau (qui est un triangle isocèle), la valeur de G est proportionnelle à la surface de celui-ci. Néanmoins, afin d'accélérer la reconnaissance de cette valeur, nous avons adopté une politique de double codage : l'indice G sera aussi associé à la couleur du drapeau. Plus le bleu est foncé plus l'activité est intense, plus il est clair plus l'activité est faible.

Notons qu'afin d'éviter que le drapeau ne prenne des dimensions trop importantes ou que les couleurs soient positionnées dans des gammes peu optimales en terme de perception (p.ex. variantes de bleu très foncé), un calibrage des bornes des indices doit être effectué.

La Fig. 82 illustre trois états possibles du drapeau.

- Le premier (à gauche) indique qu'il y a eu une activité importante à la fois en quantité (Qt – longueur) et en motivation coopérative ($N-Qual$ – hauteur). L'activité globale était importante (G – surface et couleur).
- Par rapport au premier, le second (au centre) montre une diminution en termes de motivation coopérative ($N-Qual$).
- Le troisième drapeau est associé à une situation de faible activité coopérative à la fois au niveau quantitatif et qualitatif.

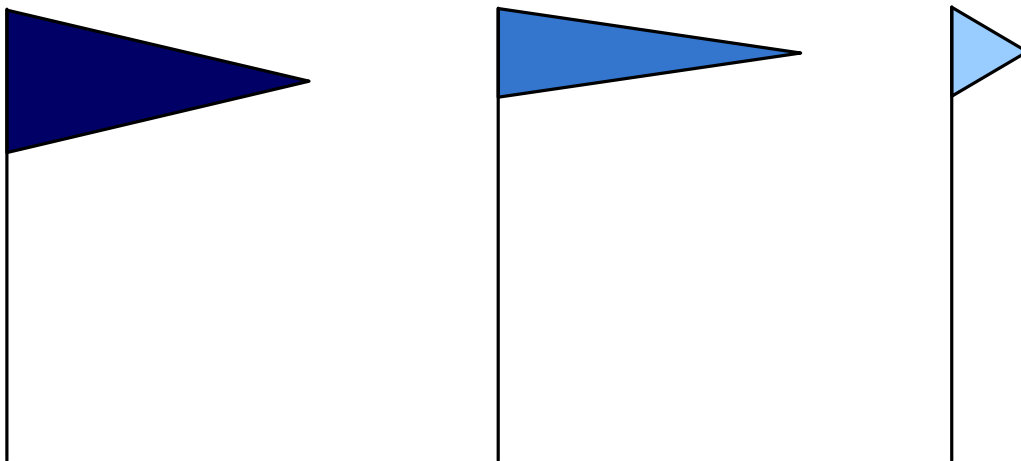


Fig. 82: Exemple de drapeaux associés à des niveaux différents d'activité

²⁴ La hauteur du drapeau réfère à celle de la pièce de tissu et non à la position sur le mât.

5. Visualisation 'ClockView'

5.1 Contexte de travail

Dans les sections précédentes de ce chapitre, nous avons discuté de la problématique des indices d'activité coopérative. Nous avons défini certains indicateurs et nous en avons proposé une visualisation. Ces indices sont utiles pour fournir une vue globale d'une situation de coopération électronique. Par contre, ils ne donnent aucune information sur les interactions individuelles. Il est cependant évident que ce niveau de granularité est également utile dans certaines circonstances. Cette question sera abordée dans la première partie de ce chapitre.

Nous avons identifié dans notre taxonomie IVF un ensemble de propriétés permettant de caractériser une technique de visualisation d'activités coopératives. Il nous semble important d'attirer ici l'attention sur l'une d'entre elles : la propriété de '*centricité*'. Elle est importante dans le contexte de la représentation d'interactions individuelles. Elle indique si les interactions à visualiser se réfèrent à une entité particulière (vue égocentrique) ou concernent un ensemble d'entités sans en favoriser une en particulier (vue sociocentrique). Dans un premier temps, nous avons choisi de nous focaliser sur la conception d'une vue égocentrique. En effet, bien que nous étudions des situations qui impliquent de la coopération au sein d'un groupe, il est important de permettre aux acteurs individuels de connaître l'état de leurs interactions propres.

Nous avons montré dans notre état de l'art que de nombreuses techniques ont été proposées pour représenter les interactions électroniques d'une personne donnée. Une des limites habituellement rencontrées concerne l'intégration dans une même vue de données issues de technologies différentes supportant une même fonctionnalité (p.ex. deux clients de messagerie distincts) ou issues de systèmes offrant des fonctionnalités différentes (p.ex. un client de messagerie et un système d'agenda). Nous avons donc décidé d'essayer de proposer une solution à cette problématique.

De plus, nous avons voulu mettre en évidence le côté temporel des interactions. Cependant, alors que la plupart des solutions proposées jusqu'à présent illustrent la linéarité du temps, il nous a semblé utile d'explorer sa dimension cyclique. L'idée sous-jacente est de proposer à l'utilisateur une vue qui lui montre l'ensemble des interactions dans lesquelles il est impliqué au cours d'une journée donnée.

5.2 Principes de conception

Notre but était donc de fournir à un utilisateur une vue d'ensemble de ses interactions relatives à un jour particulier. Cette approche égocentrique s'est matérialisée par la visualisation originale '*ClockView*'.

Plusieurs éléments expliquent la représentation graphique proposée.

Premièrement, la vue est destinée également à positionner temporellement les interactions. La granularité retenue est la journée. Dès lors, nous avons choisi d'utiliser la métaphore de l'horloge qui est universellement connue pour positionner des événements dans une journée. La visualisation proposée se compose donc de deux cercles imitant le principe d'une horloge (cf. Fig. 83 a.). Le premier cercle est relatif à la période minuit – midi (matin) et le second à la période midi – minuit (après-midi).

Deuxièmement, seules les interactions relatives à un utilisateur donné sont concernées. Il s'agit donc d'une approche de type égocentrique. Pour illustrer ce concept, nous avons adopté la métaphore d'un groupe au milieu duquel serait situé l'utilisateur de référence dont on visualise les interactions (cf. Fig. 83 b.). Les interactions initiées par lui seront représentées par des flèches dont il est l'origine et les interactions qui lui sont destinées par des flèches qui pointent vers lui. L'utilisateur de référence est donc 'virtuellement' placé au centre de l'horloge.

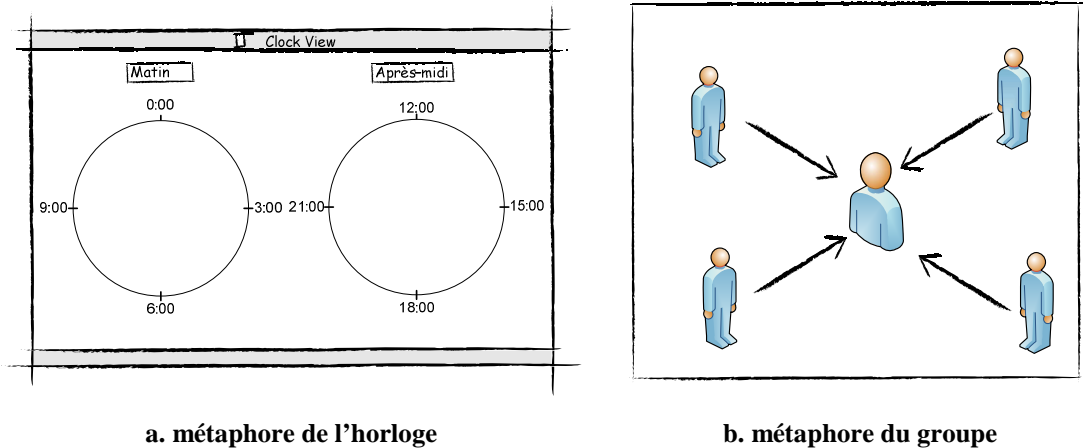


Fig. 83: Concepts à la base de la visualisation 'Clock View'

Troisièmement, nous distinguons les interactions ponctuelles et longues selon que leur durée soit significative ou non. Une interaction ponctuelle sera représentée par un rayon positionné sur un des deux cercles comme une aiguille d'horloge. Sa position angulaire correspond à l'heure associée à cette interaction. La Table 14 indique le positionnement temporel y relatif dans différents cas de figure.

Type d'interaction	Durée	Informations temporelles associées	
		Initiée par l'utilisateur	Destinées à l'utilisateur
Courriel	Interact. ponctuelle	Date ²⁵ d'envoi	Date de réception
Publication d'événement (calendrier)	Interact. ponctuelle	Date de publication	Date de lecture
Nouvelle (<i>News</i>)	Interact. ponctuelle	Date de publication	Date de lecture
Liste de diffusion (<i>Mailing list</i>)	Interact. ponctuelle	Date d'envoi	Date de réception
Document	Interact. ponctuelle	Date de chargement (<i>upload</i>)	Date de téléchargement (<i>download</i>)

Table 14: Positionnement temporel des interactions ponctuelles dans la visualisation 'ClockView'

Une interaction longue (p.ex. réunion) est visualisée par un secteur de cercle défini sur le cercle par son heure de début et son heure de fin (cf. Fig. 84).

L'intégration de ces différents éléments (cf. Fig. 84) définit conceptuellement la visualisation 'Clock View'. On remarque immédiatement l'utilisation combinée des métaphores de l'horloge et du groupe de personnes. De plus, quoiqu'originale, cette représentation est très intuitive.

²⁵ Le terme 'Date' fait référence au concept de 'timestamp'. Il est donc défini comme un positionnement temporel absolu (année, mois, jour, heure, minutes, secondes,...).

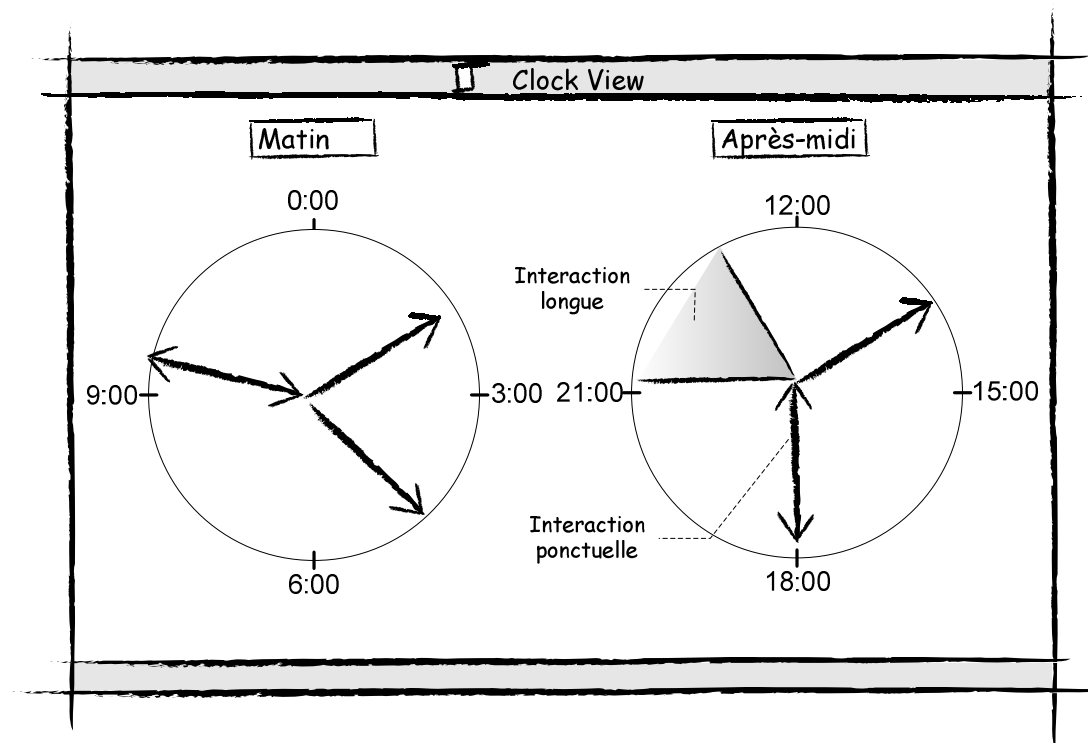


Fig. 84: Intégration des concepts sous-jacents à la visualisation 'Clock View'

5.3 Prototype

5.3.1. Architecture logicielle

L'application 'Clock View' repose sur la même architecture logicielle que celles destinées à visualiser les indices d'activité coopérative (cf. p. 164). Elle se compose de trois modules indépendants (cf. Fig. 85).

- **La plate-forme électronique collaborative** basée sur les technologies habituelles Apache, PHP, MySQL. Sur la base de données MySQL (A) ont été ajoutés des *triggers* qui sont déclenchés lorsque des interactions sont initiées. Ils alimentent une seconde base de données MySQL (B). La liste de diffusion fait appel à des fichiers 'texte' qui sont lus par un module dédié écrit en langage PHP. Ce dernier insère la base de données B les interactions correspondantes.
- **Une base de données B** dans laquelle est implémenté notre modèle de la coopération basé sur des entités et des interactions.
- **Un module de visualisation** des interactions écrit en langage Java. Il interroge la base de données B pour lire les interactions à afficher.

Cette architecture est extensible pour intégrer des données issues de technologies hétérogènes. En effet, l'introduction d'une couche intermédiaire (c.-à-d. la base de données B) qui implémente notre modèle de la coopération permet aux divers modules de visualisation de ne pas être dépendant des technologies qui supportent concrètement les interactions électroniques. Dans notre cas, nous avons ainsi combiné des interactions reposant sur l'utilisation de la base de données A (p.ex. partage de documents) et d'autres utilisant des fichiers 'texte' (liste de diffusion). Si nous souhaitons intégrer dans le système des interactions liées, par exemple, à un client de messagerie complémentaire, il suffit d'alimenter automatiquement la base de données B avec les informations utiles sur les courriels échangés par ce biais.

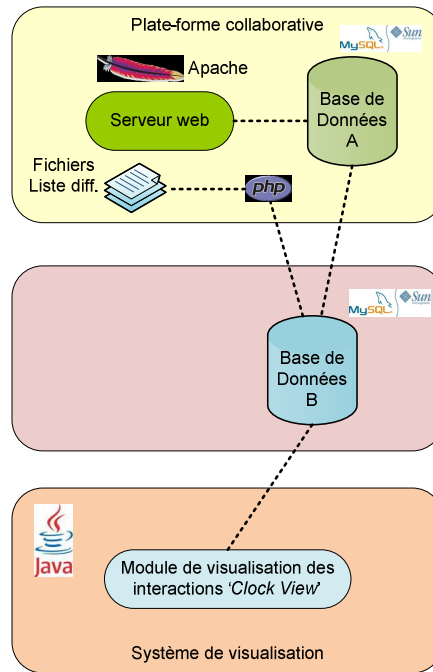


Fig. 85: Architecture informatique du système de visualisation 'Clock View'

5.3.2. Interface utilisateur et fonctionnalités

D'une manière globale, l'interface utilisateur se décompose en trois grandes parties (cf. Fig. 86). La zone supérieure de la fenêtre est destinée à sélectionner les informations à afficher. La zone centrale visualise les informations sélectionnées. La zone inférieure permet d'afficher des détails.

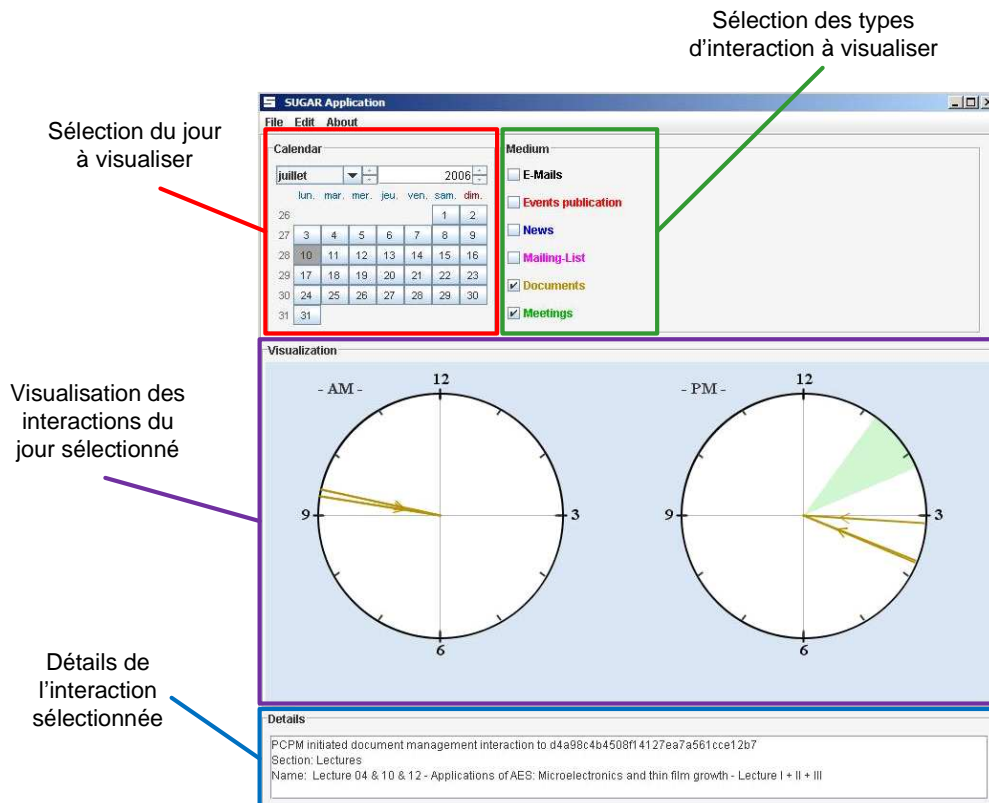


Fig. 86: ClockView : organisation de l'interface utilisateur

Les fonctions habituelles pour une telle interface utilisateur ont été intégrées dans le prototype.

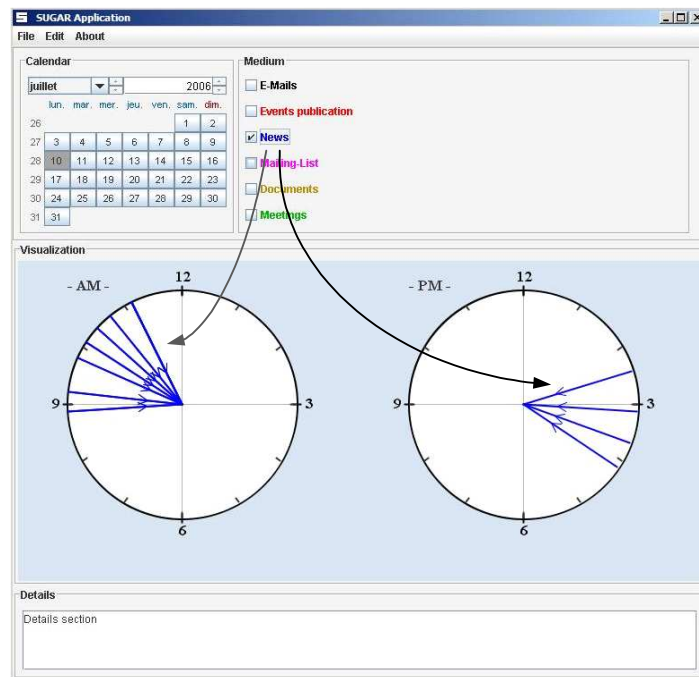


Fig. 87: Sélection des types d'interaction à visualiser

La **sélection des interactions** à afficher peut être opérée sur base de deux critères :

- la date, sélectionnable à l'aide d'un calendrier,
- le type d'interaction, choisi à l'aide de cases à cocher (*check boxes*).

Chaque type d'interaction est associé à une couleur particulière, ce qui permet de les distinguer facilement. Par exemple, dans la Fig. 87, l'utilisateur a choisi de visualiser uniquement les nouvelles (*news*).

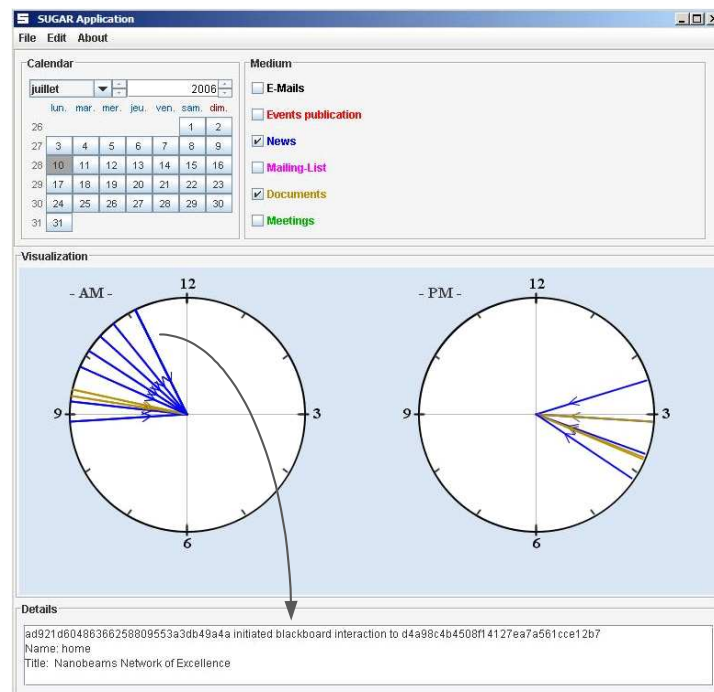


Fig. 88: Affichage du détail de l'interaction sélectionnée

Une fonction '**Vue d'ensemble – Détail**' (*Overview – Detail*) est également présente. Quand l'utilisateur sélectionne une interaction dans la zone 'horloge', des informations complémentaires sur cette interaction sont affichées dans la zone 'texte' située dans la partie inférieure de la fenêtre (cf. Fig. 88).

Comme nous l'avons discuté précédemment, la directionnalité des interactions est indiquée par le sens des flèches associées à celles-ci.

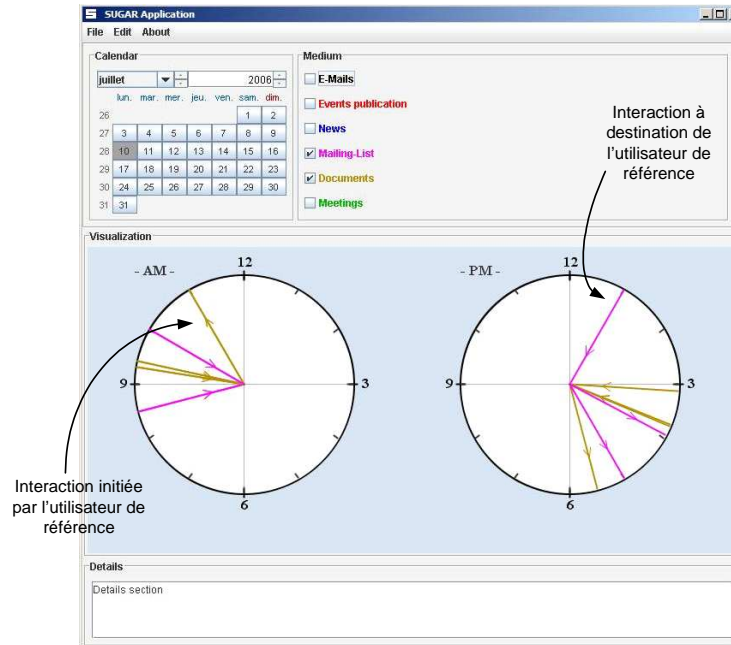


Fig. 89: Directionnalité des interactions

Notons que pour les réunions, nous n'avons pas estimé utile d'indiquer une directionnalité. Le concept d'envoi – réception est en effet moins applicable aux réunions qu'aux autres types d'interaction.

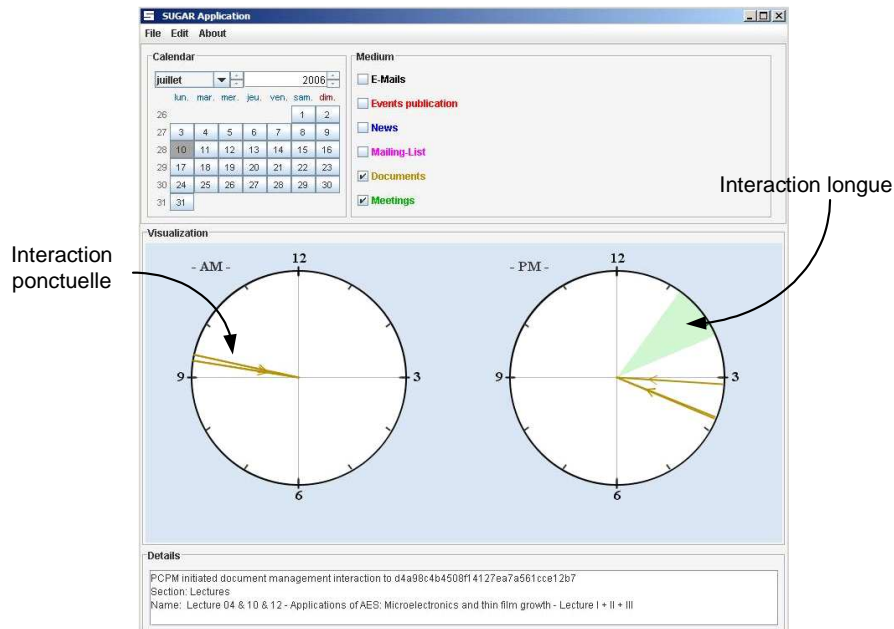


Fig. 90: Représentation des interactions ponctuelles et longues

Enfin, la durée des interactions se reflète dans la surface du secteur de cercle correspondant. Dans le cas d'interactions ponctuelles, le secteur de cercle se réduit à une droite.

Chapitre 7 : Ellimaps et dérivés

« Il y a dans les sciences plusieurs places honorables, comme il y a, si on en croit l'Évangile, plusieurs demeures dans la maison du Père céleste. »

J. d'Alembert

1. Contexte

Nous disposons à présent de différents moyens de visualiser des informations relatives à une plate-forme électronique de coopération.

- une visualisation de l'évolution temporelle des indices d'activité (cf. Fig. 78),
- une visualisation périphérique du niveau courant d'activité (cf. Fig. 79),
- une visualisation égocentrique des interactions journalières (cf. Fig. 86).

Les deux premières techniques sont utiles pour connaître rapidement l'état global de la coopération ou son évolution temporelle. Elles ne fournissent par contre aucune information sur les interactions individuelles.

La troisième technique est intéressante pour un utilisateur pour positionner temporellement les interactions qui le concernent pour un jour donné mais elle ne permet pas de visualiser des périodes de temps plus longues.

Nous avons donc essayé de concevoir une technique de visualisation complémentaire qui pourrait permettre de visualiser des éléments d'information ayant une granularité plus fine que les indices d'activité et qui pourrait porter sur des périodes de temps plus longues qu'une journée.

Au cours de notre réflexion, nous nous sommes rendu compte qu'une approche générique représentait un choix judicieux pour répondre à cette question. Au lieu d'imaginer une nouvelle technique qui intègre par conception la nature des données à visualiser²⁶, nous avons décidé d'explorer la piste d'une technique générique qui pourrait s'adapter facilement à des données sémantiquement différentes.

En fait, alors que les techniques 'Ambient Flag' et 'Clock View' étaient dérivées de la sémantique des données à représenter, cette nouvelle approche prit pour base la structure de celles-ci.

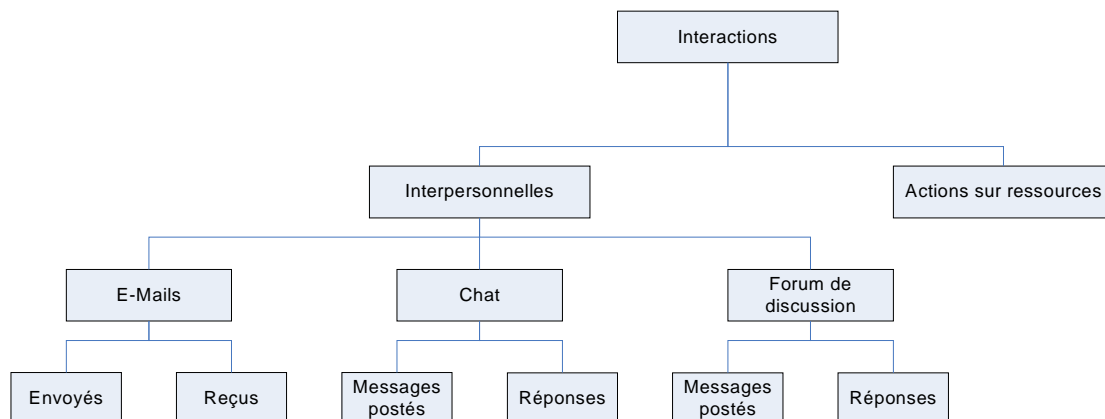


Fig. 91: Exemple de structuration hiérarchique des interactions

²⁶ Ce qui est le cas des visualisations 'Clock View' et 'Ambient Flag'

Plus précisément, nous avons pris conscience du fait que la structuration hiérarchique des données est largement présente dans les systèmes collaboratifs. Quelques exemples suffisent à illustrer ce propos. L'organigramme de personnes au sein d'une organisation, la décomposition d'un projet en tâches et sous-tâches, la structuration d'un système de partage de documents ou une taxonomie de mots clés catégorisant des messages peuvent tous être modélisés comme des données hiérarchiques.

Nous pouvons également ajouter que d'autres données qui, à première vue, n'apparaissent pas regroupées sous forme hiérarchique peuvent le devenir après un traitement assez simple :

- le temps décomposé sous la forme : mois – jour – heure – minute,
- les interactions structurées au sein d'un groupe selon certaines de leurs propriétés (cf. Fig. 91)

Il nous est donc apparu utile d'approfondir la question de la visualisation de données hiérarchiques dans le contexte des systèmes collaboratifs.

2. Visualisation de données hiérarchiques

La visualisation de données hiérarchiques est bien documentée dans la littérature. Les nombreuses techniques qui ont été proposées dans ce domaine se fondent essentiellement sur deux propriétés visuelles : la *clôture* et la *connectivité*.

La *connectivité* est de loin la plus utilisée. Pour beaucoup de personnes, les arbres verticaux classiques (2-D), dans lesquels l'axe vertical est associé à la profondeur de l'arbre et l'axe horizontal est utilisé pour séparer les nœuds apparaît comme le moyen le plus simple et le plus intuitif pour représenter des données hiérarchiques (cf. Fig. 91). Malheureusement, ce type de visualisation devient rapidement inefficace à mesure que croît le nombre de nœuds à afficher. De plus, ces représentations contiennent souvent beaucoup d'espace vide, ce qui les pénalise en termes d'occupation de l'espace d'affichage (Card et al., 1999, p. 149). Parmi les stratégies visant à proposer des alternatives, l'utilisation de coordonnées non cartésiennes et de géométries spéciales (c.-à-d. non Euclidiennes) constitue une approche assez répandue. Par exemple, si des coordonnées polaires sont utilisées, un arbre circulaire (*disk tree*) (Chi et al., 1998) peut être construit en mettant en correspondance la profondeur de l'arbre avec le rayon du cercle et en utilisant la distance angulaire pour distinguer les nœuds. Un autre exemple est fourni par le navigateur de Lamping et Rao (1994) qui utilise la géométrie hyperbolique. La technique du '*Cone Tree*' (Robertson et al., 1991) est encore une autre approche qui visualise les descendants d'un nœud sous la forme d'un cône 3-D qui émerge du nœud parent.

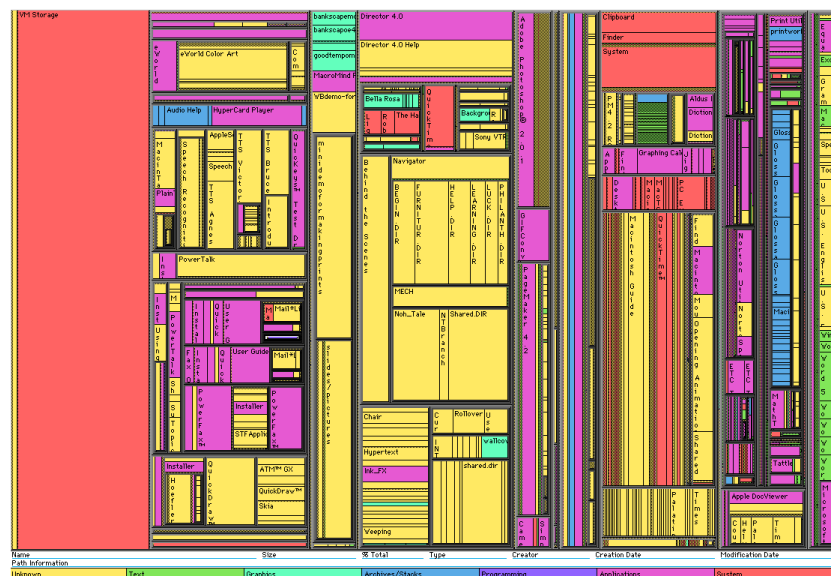


Fig. 92: Treemaps (source: [182])

La *clôture* est une seconde propriété visuelle utilisée pour visualiser des données hiérarchiques. Elle consiste à inclure des formes géométriques les unes dans les autres. Notons que certaines de ces techniques sont parfois nommées techniques de remplissage d'espace (*space-filling methods*). A ce niveau, il est important de mentionner que, dans la plupart des cas, ces techniques se différencient des approches basées sur la connectivité en intégrant dans leur nature propre la visualisation d'une valeur numérique affectée aux nœuds (cf. nœuds pondérés). Les *treemaps* conçus par Shneiderman (1992) au début des années 1990 constituent probablement la plus célèbre technique basée sur la propriété de clôture pour visualiser des hiérarchies (cf. Shneiderman, 2006 pour une revue historique de l'approche *treemap* et ses nombreuses évolutions). Les *treemaps* ont été notamment utilisés pour visualiser la structure des répertoires d'un disque dur d'ordinateur (Johnson et Shneiderman, 1991) (cf. Fig. 92) ou l'évolution des marchés financiers (Smartmoney, 2006).

Le principal avantage des *treemaps* réside dans le fait que l'espace d'affichage nécessaire reste constant quelles que soient la largeur et la profondeur de la structure hiérarchique. De plus, l'aire de chaque rectangle est proportionnelle à la valeur numérique associée au nœud correspondant. Cette propriété facilite la perception de la taille relative ou de l'importance de chaque nœud. Par exemple, ils ont été utilisés pour visualiser des résultats électoraux par circonscription (cf. Fig. 93), le nombre de messages associés à des groupes de discussion (cf. Fig. 94) ou le nombre de nouvelles sur un sujet donné dans *Google News* (cf. *Newsmap*, Fig. 95).

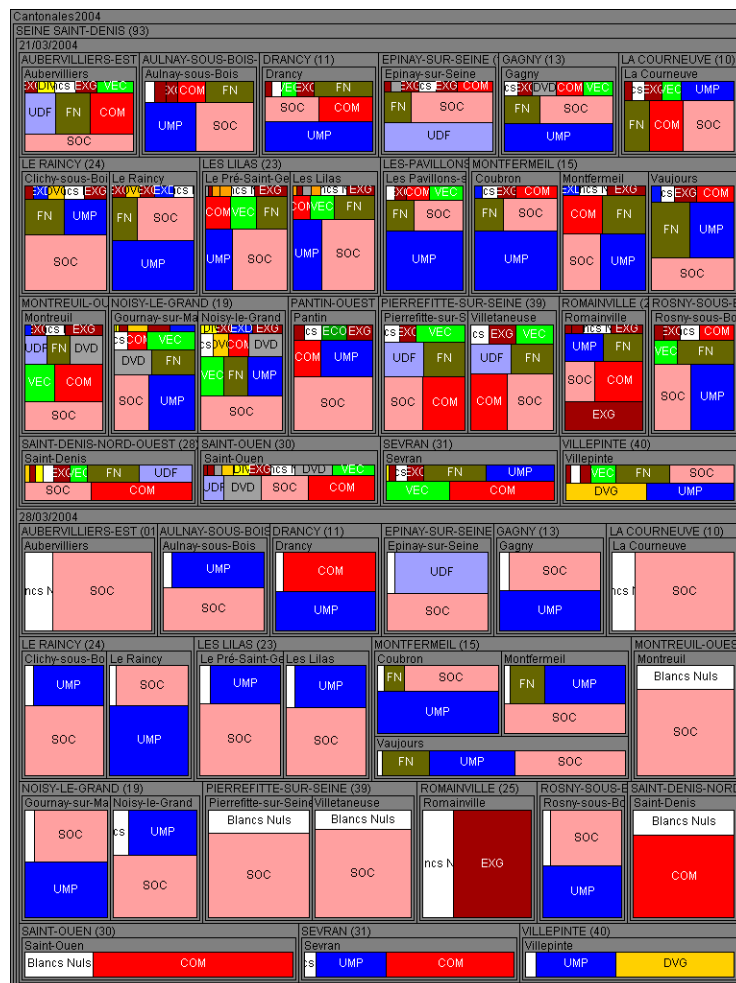


Fig. 93: Visualisation de résultats électoraux (source: [65])

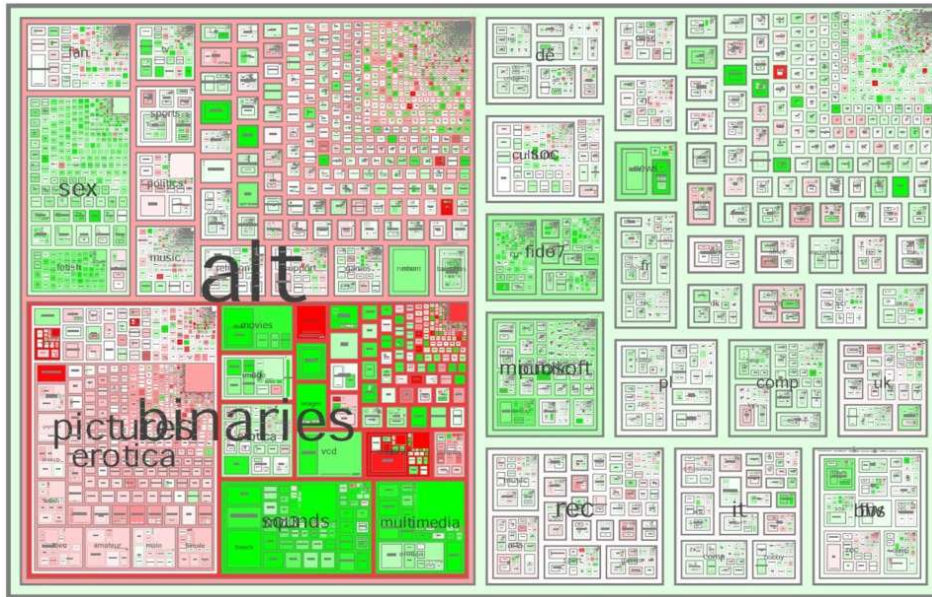


Fig. 94: Visualisation de messages dans *Usenet* (source: [187])



Fig. 95: Visualisation du nombre de nouvelles par sujet (source: [217])

Malheureusement, les *treemaps* présentent aussi des limitations. Pour certaines distributions des nœuds dans la hiérarchie et des valeurs qui y sont associées, le *treemap* peut devenir difficile à comprendre (p.ex. les rectangles deviennent des lignes). La technique ‘*squarified treemaps*’ (Bruls et al., 2000) a été proposée pour répondre à cette faiblesse (cf. p. 195). Elle essaie de conserver aux rectangles un rapport longueur / largeur proche de celui d’un carré.

Par ailleurs, les *treemaps* ne sont pas optimaux pour représenter l’inclusion des sous-hiérarchies les unes dans les autres. Comme le mentionne Ware (2004, p. 217), ‘le désavantage [des *treemaps*] est que la structure hiérarchique n’est pas aussi claire qu’elle l’est dans une représentation d’arbre conventionnelle, qui est une forme spéciale de diagramme nœud-lien.’ Winckler et al. (2004) rejoignent cette position : ‘bien qu’il [*treemap*] représente des attributs importants, il ne communique pas la structure aussi bien que les diagrammes nœuds – liens traditionnels’.

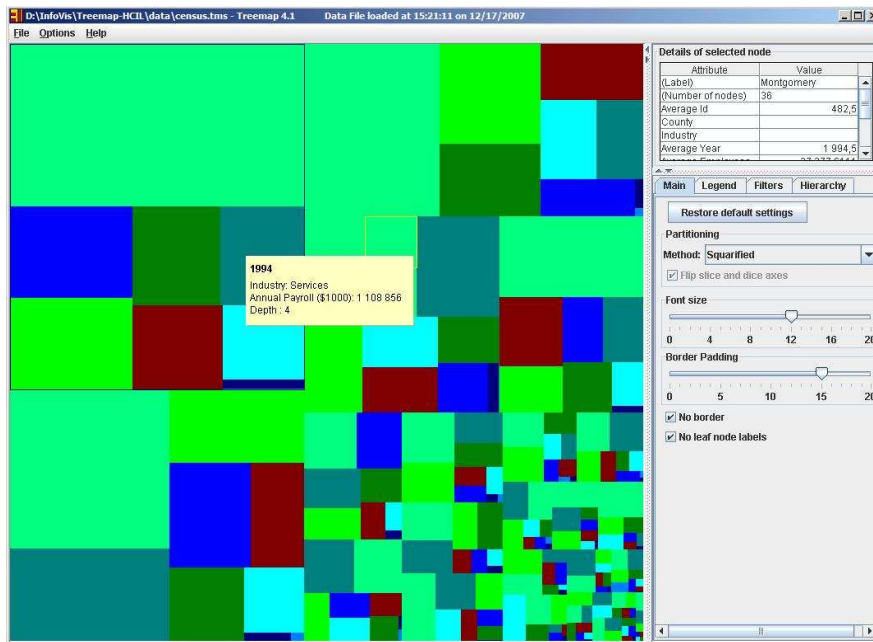


Fig. 96: Treemap simple (source: [89])

Cette limitation s'applique par exemple au treemap simple (cf. Fig. 96). Il apparaît assez évident que dans ce cas, la perception de la structure de la hiérarchie n'est pas aisée si aucun autre élément n'est ajouté pour la représenter.

Cette limitation des *Treemaps* a suscité des recherches visant à rendre plus explicite la hiérarchie de l'arbre représenté. L'ajout d'un effet de bordure ('padding effect') est une première réponse (cf. Fig. 97). Elle permet en outre d'améliorer l'étiquetage des nœuds.

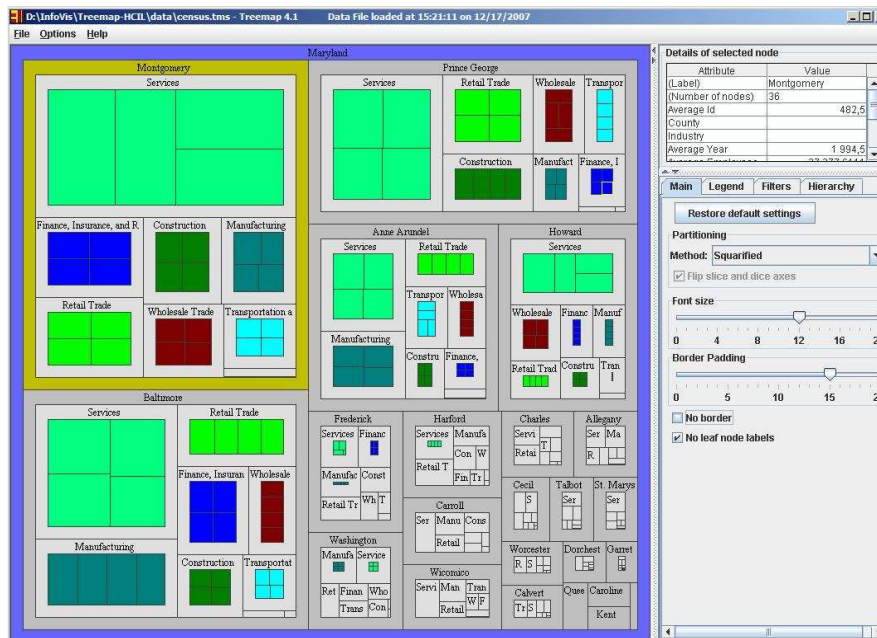


Fig. 97: Treemap avec effet de bordure (source: [89])

Une autre approche est fournie par la variante '*Cushion TreeMap*' (van Wijk et van de Wetering, 1999) qui utilise des algorithmes d'ombrage (cf. Fig. 98). Elle a été évaluée empiriquement par Irani et al. (2006) qui ont conclu que l'ombrage facilitait la perception de la structure arborescente.

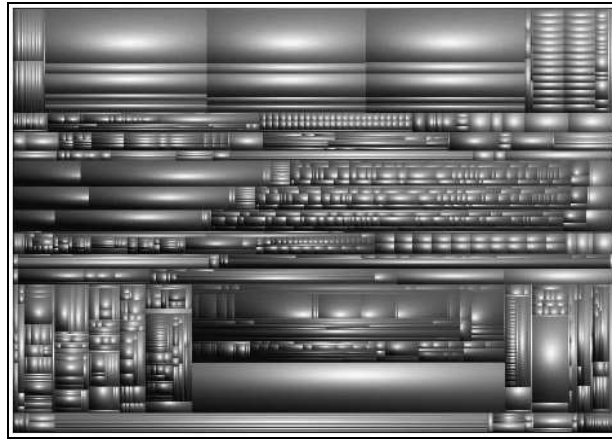


Fig. 98: Cushion Treemap (source : [223])

Tout comme les approches basées sur la connectivité, les techniques basées sur la loi de clôture ont été adaptées à des systèmes de coordonnées et à des géométries moins habituelles. Par exemple, l'approche '*pietree*' (cf. Dix et al., 2000 ; O'Donnel et al., 2006) se base sur un système de coordonnées polaires et peut être considérée comme le pendant de la technique '*disk tree*'.



Fig. 99: Voronoi Treemap (source : [8])

La technique '*Voronoi treemaps*' (Balzer et al., 2005) représente un autre exemple d'adaptation du concept de *treemap*. Elle repose sur le choix de polygones au lieu de rectangles comme primitives géométriques.

Vliegen et al. (2006) ont pour leur part appliqué des déformations pour changer l'aspect de *treemaps*. Les transformations sont particulières car elles conservent le ratio des surfaces. Leur objectif était de faire ressembler les *treemaps* aux diagrammes habituellement rencontrés pour visualiser des données de gestion.

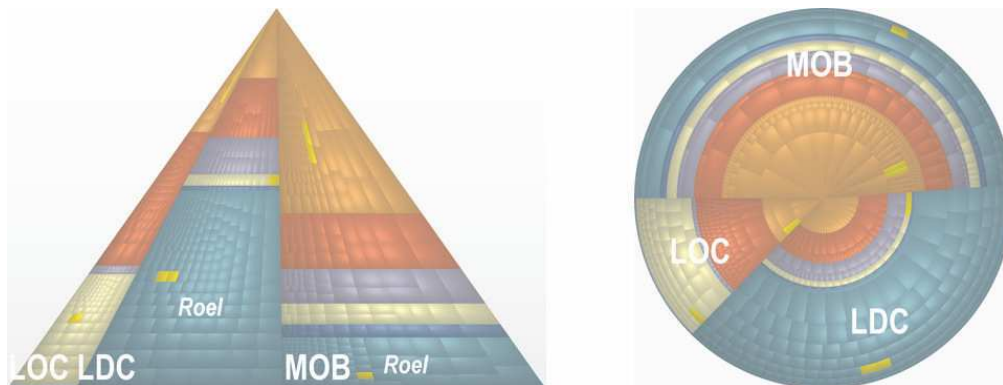


Fig. 100: Déformation de treemaps (source : [210])

Dans les deux groupes de visualisations de données hiérarchiques, des techniques d'interaction avancées sont habituellement utilisées pour améliorer la puissance et la convivialité du logiciel. Par exemple, les techniques telles que le '*focus+context*', l'œil de poisson (*fish-eye view*) ou les *queries* dynamiques (*dynamic queries*) ont été utilisées à cet effet. Très récemment, constatant le moindre intérêt porté dans la littérature à l'interaction spécifique avec des *treemaps*, Blanch et Lecolinet (2007) ont proposé de combiner les concepts d'interface utilisateur zoomable (*Zoomable User Interface*) et de *treemap* pour donner naissance à ce qu'ils ont nommé Treemap zoomable (*Zoomable Treemap*). Cette technique introduit notamment des animations déclenchées au cours de la navigation dans le *treemap*, ce qui rend celle-ci plus progressive que l'approche habituelle basée sur des transitions instantanées d'un nœud à l'autre.

Pour conclure, on peut retenir que, dans les techniques de visualisation de données hiérarchiques, un défi majeur consiste à atteindre un équilibre optimal entre la perception aisée de la structure arborescente, le taux d'occupation de l'espace d'affichage et la représentation graphique de la valeur associée aux nœuds. La faiblesse de la plupart des techniques basées sur la loi de clôture en termes de reconnaissance de la structure des données est contrebalancée par leur puissance pour utiliser efficacement l'espace d'affichage et pour représenter la valeur numérique des nœuds. De manière symétrique, les techniques reposant sur la connectivité s'avèrent très puissantes pour mettre en évidence la structure hiérarchique mais pêchent lorsque l'occupation de l'espace ou la représentation de la valeur des nœuds est prise en considération.

Nous avons choisi de nous concentrer sur les techniques basées sur la loi de clôture. En toute logique, nous avons essayé de répondre à une de leurs faiblesses : la perception de la structure des données. Nous étudierons donc la question suivante : « Comment concevoir une technique de visualisation basée sur la propriété de clôture qui offre une perception intuitive de la structure hiérarchique des données ? »

Le concept que nous proposons se base sur une observation très simple de la vie réelle. Si on demande à une personne d'imaginer une figure représentant des groupes d'éléments successivement agrégés les uns dans les autres (c.-à-d. des ensembles et des sous-ensembles d'objets), elle va dessiner, dans un grand nombre de cas, des pseudo-ellipses imbriquées les unes dans les autres²⁷. En fait, ce type de données est typiquement structuré comme une hiérarchie. De plus, naturellement, les ensembles contenant plus d'éléments seront associés à des ellipses de plus grande taille. Si on établit un parallèle entre cette observation et le problème scientifique que nous avons posé, l'idée de remplacer les rectangles par des ellipses dans une approche de type *treemap* apparaît assez naturelle. Nous appellerons cette nouvelle technique de visualisation : ellimap.

²⁷ Au point de vue de la psychologie cognitive, il serait intéressant d'analyser si ce comportement est inné ou acquis. Une telle étude pourrait, par exemple, essayer de déterminer l'influence de la formation en mathématiques (cf. théorie des ensembles) dans ce comportement.

Une revue de la littérature nous apprend que deux chercheurs ont précédemment exploré des idées qui, dans une certaine mesure, montrent des similarités avec le concept de base des ellimaps : Boardman avec son travail sur les *Bubble Trees* (Boardman, 2000) et Wetzel (2004) avec sa technique *Circular Treemaps*.

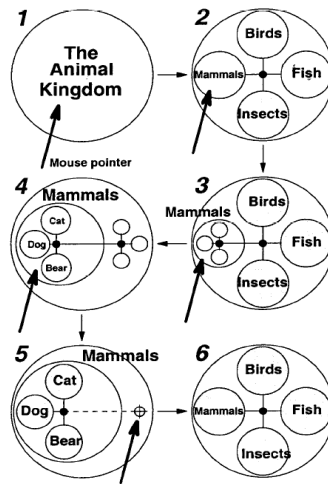


Fig. 101: Technique ‘Bubble Trees’ (source: [26])

Boardman a imaginé un moyen de visualiser des structures arborescentes qui repose sur leur propriété naturelle d’être divisibles en sous-arbres (cf. Fig. 101). Dans les *Bubble Trees*, chaque sous-arbre est représenté par un cercle inscrit dans un autre cercle associé au niveau hiérarchique supérieur. L’inclusion successive des niveaux hiérarchiques est clairement perceptible. Par contre, la valeur numérique associée aux nœuds n’est pas représentée.

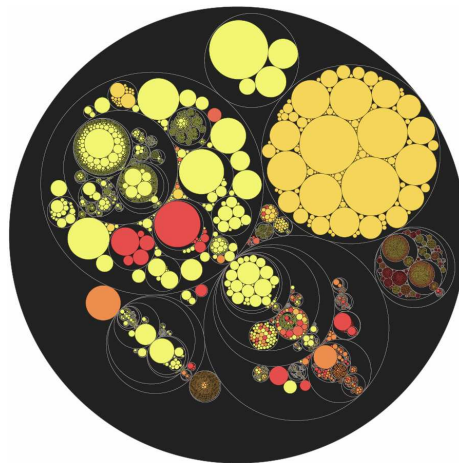


Fig. 102: Visualisation ‘Circular Treemaps’ (source: [218])

L’approche *Circular Treemap* (cf. Fig. 102) est une évolution du concept de *treemap* qui s’en différencie en adoptant le cercle au lieu du rectangle comme primitive graphique. Cette technique a été utilisée pour visualiser la hiérarchie des répertoires sur un disque dur d’ordinateur. Comme dans le cas des *Bubble Trees*, la perception de l’inclusion successive des niveaux hiérarchiques est aussi très intuitive. Shneiderman (2006) reconnaît cet avantage mais il ajoute, de concert avec Wetzel, que c’est au détriment de l’efficacité d’occupation de l’espace (*‘Circular treemaps make for a visually attractive layout and possibly useful idea because nesting is nicely visible but the wasted space undermines its efficacy’*). De plus, Wetzel lui-même mentionne que des cercles de même surface peuvent potentiellement représenter des fichiers ou des répertoires de tailles différentes. La représentativité de l’aire

des primitives graphiques utilisées est donc perdue alors qu'elle constitue un des avantages majeurs des *treemaps*.

Malgré ces lacunes, les deux approches mettent en avant le potentiel des formes géométriques curvilignes pour représenter des ensembles imbriqués d'éléments.

En résumé, nous chercherons donc à concevoir une nouvelle technique de visualisation qui répond aux contraintes suivantes :

- technique de visualisation destinée à représenter des données hiérarchiques,
- technique qui met en évidence de manière intuitive la relation d'inclusion entre les branches de la hiérarchie,
- technique qui utilise un espace d'affichage fixe quelle que soit la structure de l'arbre,
- technique récursive dérivée de l'approche *treemaps* mais qui utilise des primitives géométriques curvilignes plutôt que des rectangles.

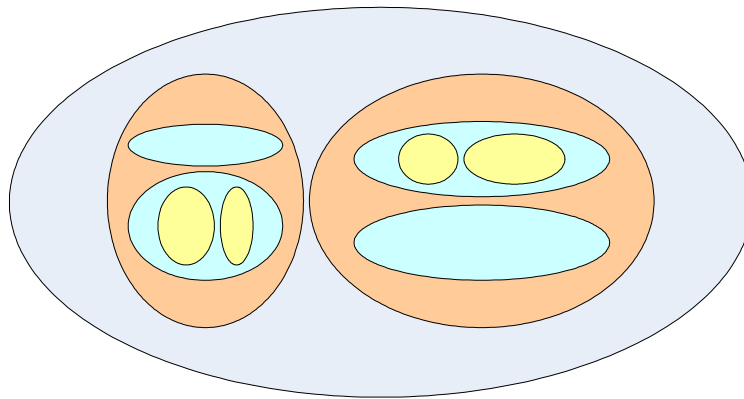


Fig. 103: Ellimaps: exemple de principe

Comme nous l'avons précisé précédemment, nous avons donc eu l'idée d'utiliser les algorithmes connus de *treemaps* mais avec une autre primitive graphique : l'ellipse (cf. Fig. 103). Très logiquement, nous avons nommé cette technique : '*Ellimaps*'.

L'ellipse présente l'avantage d'être mentalement associée à la théorie des ensembles que la plupart des personnes connaissent, au moins dans ses concepts rudimentaires. La majorité des personnes utilisent d'ailleurs intuitivement des ellipses pour représenter graphiquement des ensembles d'éléments. Cette primitive graphique s'avère donc être un choix assez naturel pour représenter des ensembles d'éléments imbriqués.

Des considérations théoriques plaident également en faveur de l'exploration du potentiel des ellipses pour remplacer les rectangles dans une approche de type '*treemaps*'.

Nous nous sommes particulièrement intéressés à la théorie de la '*Gestalt*' élaborée par Westheimer, Koffka and Kohler au début du 20^{ème} siècle. Ware (2004, p. 189) explique que bien que les explications neurologiques proposées à l'époque n'aient pas réussi à passer l'épreuve du temps, les lois elles-mêmes ont prouvé qu'elles étaient durables. D'un point de vue général, cette théorie explique que la perception des objets suit des lois naturelles qui commandent l'attention du sujet. Sa loi fondamentale est la loi de *Prägnanz* qui dit que l'esprit humain a tendance à percevoir en premier lieu des 'bonnes formes', c.-à-d. des formes régulières, ordonnées, simples ou symétriques. Plusieurs autres lois font partie de la théorie de la '*Gestalt*'. Nous nous focaliserons sur quatre d'entre elles qui présentent de l'intérêt dans le contexte de notre questionnement : la loi de clôture (*closure*), la loi de continuité (*continuity*), la loi de similarité (*similarity*) et la loi de taille relative (*relative size*). Nous incluons également dans notre réflexion les lois de région commune (*common region*) et de connectivité (*connectedness*) qui ne figurait pas dans la théorie initiale mais qui ont été proposées plus tard par Palmer (1992) et Palmer et Rock (1994) comme des principes de '*Gestalt*'.

La loi de clôture explique que nous avons une tendance perceptuelle à clore les contours qui présentent des interruptions. Par ailleurs, Palmer (1992) a montré qu'une région délimitée par un contour est un principe d'organisation plus fort que la simple proximité (loi de région commune, Ware, 2003). Quand nous voyons un contour fermé, nous avons une forte tendance à diviser l'espace en deux sous-espaces : 'à l'intérieur' et 'à l'extérieur' du contour. Les différentes versions de *treemaps*, les *Bubble Trees* de même que les *Circular Treemaps* reposent tous sur la loi de région commune pour amener l'utilisateur à percevoir la structure imbriquée des données qu'elles visualisent.

La loi de continuité précise que '*il est plus probable que nous construisions des entités visuelles à partir d'éléments qui sont lisses et continus qu'à partir d'autres qui présentent des changements abrupts de direction*' (Ware, 2004, p. 191). Selon ce principe, des primitives géométriques telles que les cercles et les ellipses constituent un meilleur choix que les rectangles qui contiennent quatre angles vifs. Cette loi pourrait peut-être expliquer la confusion parfois observée en certaines circonstances pour comprendre les *treemaps*.

La loi de similarité explique que '*des éléments similaires ont tendance à être regroupés ensemble*' (Ware, 2004, p. 190). Sur cette base, il apparaît que les *treemaps*, qui sont construits comme un ensemble de rectangles imbriqués, ont une probabilité plus grande d'être confondus avec d'autres éléments rectangulaires de la visualisation (p.ex. légende, boutons) que d'autres techniques reposant sur des primitives graphiques non-rectangulaires (p.ex. cercles, ellipses). Cependant, l'influence de cette loi est probablement moindre que celle des autres car la loi de similarité ne considère pas seulement la forme mais aussi la couleur, la taille ou la brillance des objets.

La *loi de taille relative* déclare qu'en général les composants graphiques les plus petits ont tendance à être perçus comme des objets. Dans le cas où les feuilles de l'arbre des données sont des objets spécifiques (p.ex. messages, documents, personnes) et les branches supérieures sont des niveaux d'agrégation de ces objets, la loi de taille relative nous incite à représenter les feuilles avec des composants graphiques plus petits. De plus, la loi de similarité nous conseille d'utiliser des objets graphiques différents d'ellipses pour visualiser une rupture conceptuelle dans la nature des données représentées.

Enfin, la loi de connectivité affirme que '*connecter différents objets graphiques par des lignes est un moyen puissant pour exprimer qu'il existe une relation entre eux*' (Ware, 2004, p. 191). Bien que cette loi ne soit pas utilisée dans les techniques de remplissage d'espace telles que celles discutées précédemment, nous pensons utile de la mentionner car il s'agit du principe fondateur de l'autre classe de visualisation de données hiérarchiques (cf. *vertical tree*, *disk tree*, *hyperbolic browser*, *cone tree*...).

En résumé, plusieurs arguments théoriques plaident pour investiguer le potentiel que pourrait présenter la substitution des rectangles par des ellipses dans une approche de type *treemap*. A ce stade, nous pouvons donc reformuler notre question scientifique. Elle consiste à concevoir une technique de visualisation de données hiérarchiques qui se base sur l'algorithme de construction des *treemaps* mais qui utilise l'ellipse au lieu du rectangle comme primitive géométrique. De plus, nous imposons que cette technique représente la valeur numérique associée aux nœuds de l'arborescence, ce qui la rend différente des approches : *Bubble Trees* et *Circular Treemaps*.

3. Ellimaps : version de base

3.1 Principes constructifs

D'un point de vue opérationnel, les algorithmes classiques de construction des *treemaps* peuvent servir de base à la réflexion sur les moyens de générer des ellimaps.

Nous illustrerons le processus de construction des ellimaps avec un exemple. Imaginons que nous souhaitions construire un ellimap associé à l'arbre de la Fig. 104. Le nombre entre parenthèses indiqué dans chaque noeud indique la valeur de la propriété à visualiser.

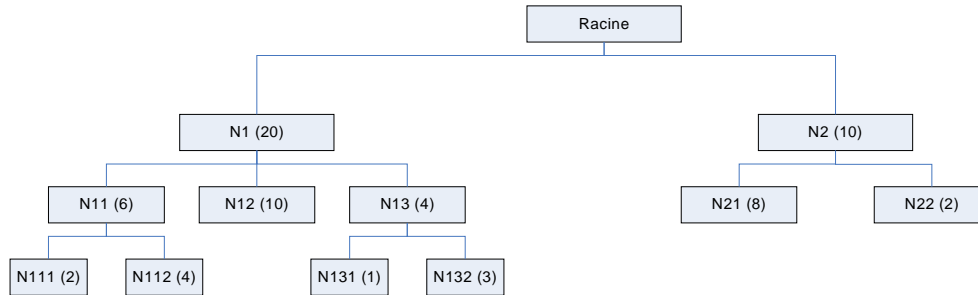


Fig. 104: Exemple de structure arborescente à visualiser

L'espace dédié à l'affichage sera représenté par un rectangle jaune (cf. Fig. 105). Il constitue l'équivalent du rectangle extérieur englobant un *treemap*. Comme notre primitive de base est une ellipse, nous remplaçons ce rectangle par l'ellipse grise qui fera office de contour extérieur de l'ellimap.

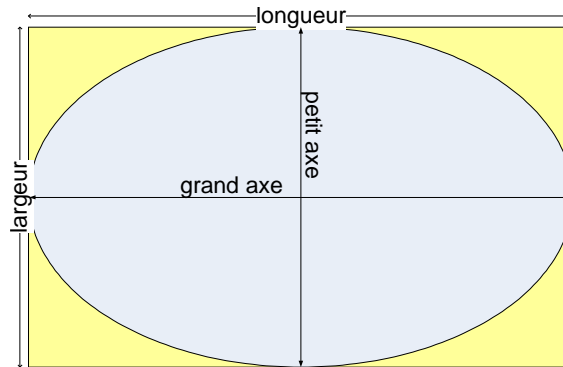


Fig. 105: Principe de l'inscription d'une ellipse dans un rectangle

Dans notre analogie entre rectangles et ellipses, le grand et le petit axe de l'ellipse seront le pendant respectivement de la longueur et de la largeur du rectangle (cf. Fig. 105).

La récursivité s'amorce par le calcul des dimensions d'un rectangle inscrit à son tour dans l'ellipse (cf. Fig. 106).

Les principes élémentaires de la géométrie nous apprennent qu'une infinité de rectangles peuvent être inscrits dans une ellipse. Par contre, un seul d'entre eux possède un rapport 'longueur/largeur' donné. Nous utiliserons ce critère pour calculer les dimensions du rectangle inscrit. Ce choix ne peut être opéré de manière purement arbitraire car il conditionne l'esthétique globale de la figure obtenue. Or, ce facteur a été reconnu comme important dans le cas des *treemaps*. En effet, l'obtention d'un ratio visuellement agréable était une des faiblesses initiales de l'approche *treemaps*. Elle a été en partie corrigée par le concept de *squarified treemaps* qui tend à générer des rectangles ayant un ratio acceptable.

Différentes stratégies sont possibles pour fixer ce ratio :

- Une première approche consiste à utiliser le ratio (a/b) calculé sur base des dimensions de l'espace d'affichage rectangulaire (rectangle jaune de longueur $2a$ et de largeur $2b$). Dans ce cas, les dimensions initiales de l'espace d'affichage conditionnent tout le processus récursif.
- Une seconde option peut être de choisir comme ratio le nombre d'or : $(1 + \sqrt{5})/2$ ($\sim 1,618033\dots$). Il s'agit d'un rapport très connu (notamment en architecture) qui est supposé être un idéal en termes esthétiques. Sa portabilité au cas des ellimaps mérite donc d'être étudiée.
- Une troisième solution vise à maximiser la surface du rectangle inscrit dans l'ellipse (de grand axe $2a$ et de petit axe $2b$). Après calcul, on obtient pour ce ratio la valeur (a/b) . En d'autres termes, cette stratégie est identique à la première.

Après divers essais, il nous est apparu que choisir le ratio $\varphi = (a/b)$ semblait être la solution la plus intéressante. Outre le fait que ce ratio tient compte des dimensions effectives de l'ellipse E , il constitue aussi celui qui maximise l'aire du rectangle R .

La Fig. 106 illustre le calcul des dimensions du rectangle inscrit.

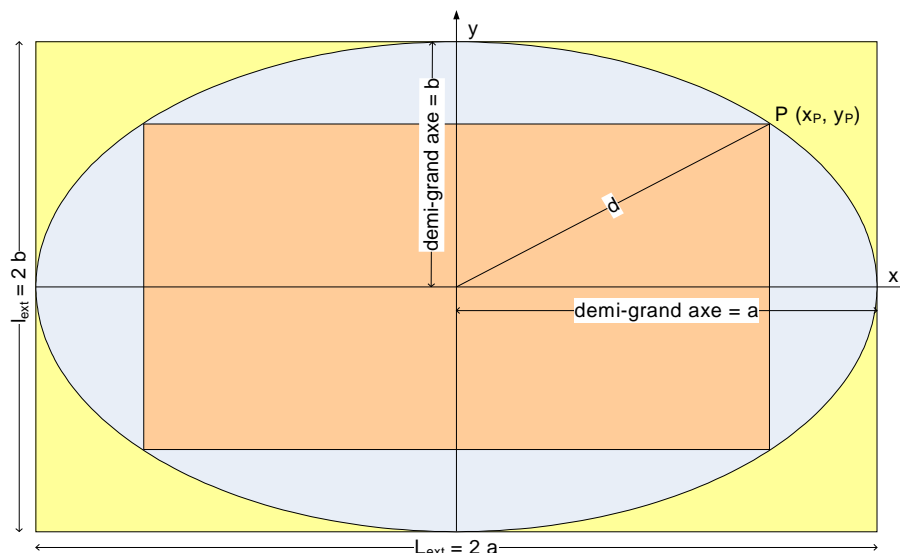


Fig. 106: Calcul du rectangle inscrit dans l'ellipse

Le choix du ratio longueur/largeur du rectangle s'exprime comme suit : $(x_P / y_P) = \varphi$ où (x_P, y_P) sont les coordonnées du point P .

En coordonnées cartésiennes, l'équation de définition d'une ellipse est : $(x^2 / a^2) + (y^2 / b^2) = 1$.

Le système formé par ces deux équations nous permet de calculer la valeur de x_P et de y_P , dès que les paramètres a , b et φ sont connus.

$$x_P = \varphi \cdot a \cdot b / \sqrt{(b^2 \cdot \varphi^2 + a^2)} \quad \text{et} \quad y_P = a \cdot b / \sqrt{(b^2 \cdot \varphi^2 + a^2)}$$

Dans un premier temps, l'algorithme classique de construction des *treemaps* sera appliqué au rectangle inscrit. Celui-ci est divisé selon l'axe horizontal en autant de rectangles que le premier niveau de l'arbre contient de branches. Dans la Fig. 106, le rectangle jaune qui circonscrit l'ellipse fait référence à la racine de l'arbre. Le rectangle orange inscrit dans l'ellipse est associé au niveau 1. Il doit être divisé en autant de rectangles qu'il y a de branches au niveau 1 de l'arbre. D'un point de vue général, s'il y a n branches au niveau 1, auxquelles sont associées les valeurs α_i , n rectangles y seront associés. Pour chacun d'entre

eux, la proportion de l'axe de division qui lui sera allouée sera calculée avec le ratio : $r_i = \alpha_i / (\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n)$.

Les coordonnées des sommets des n rectangles R_i se calculent avec l'expression :

Sommets du rectangle R_i : $\{ S_1(x_{i1}, y_P), S_2(x_{i2}, y_P), S_3(x_{i2}, -y_P), S_4(x_{i1}, -y_P) \}$ où

$$x_{i1} = -x_P + \sum_{j=1}^{(i-1)} (r_j \cdot 2x_P) \quad \text{et} \quad x_{i2} = -x_P + \sum_{j=1}^i (r_j \cdot 2x_P)$$

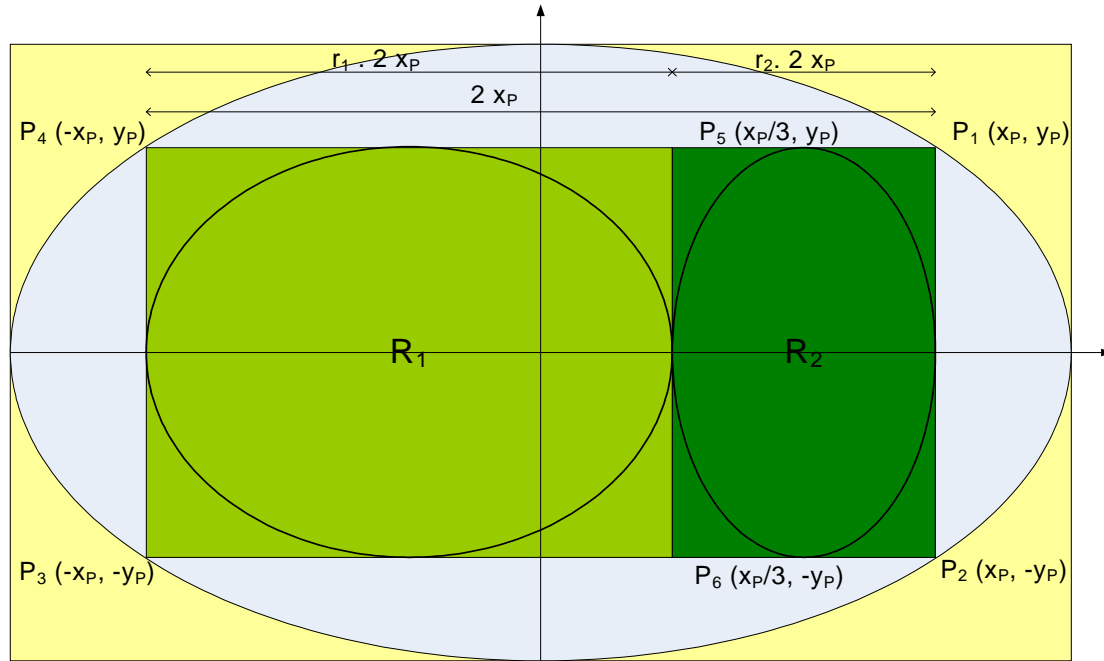


Fig. 107: Visualisation du niveau 1 de l'arbre

Dans l'exemple (cf. Fig. 107), le rectangle de niveau 1 doit être divisé en deux rectangles car il y a deux branches au niveau 1 de l'arbre. La taille des deux rectangles doit refléter le rapport entre la valeur 20 de la branche N1 et la valeur 10 de la branche N2. Nous obtenons donc les valeurs : $r_1 = 2/3$ et $r_2 = 1/3$.

Nous pouvons donc calculer les valeurs x_{i1} et x_{i2} correspondantes :

$$\begin{aligned} x_{11} &= -x_P & x_{12} &= -x_P + r_1 \cdot 2x_P = x_P/3 \\ x_{21} &= -x_P + r_1 \cdot 2x_P = x_P/3 & x_{22} &= -x_P + (r_1 + r_2) 2x_P = x_P \end{aligned}$$

Les deux rectangles ainsi définis ont pour coordonnées des sommets :

$$R_1 : \{ P_4(-x_P, y_P), P_5(x_P/3, y_P), P_6(x_P/3, -y_P), P_3(-x_P, -y_P) \}$$

$$R_2 : \{ P_5(x_P/3, y_P), P_1(x_P, y_P), P_2(x_P, -y_P), P_6(x_P/3, -y_P) \}$$

Par principe de récursivité, il faut maintenant inscrire une ellipse dans chacun des rectangles R_1 et R_2 puis calculer les coordonnées du rectangle inscrit dans ces deux ellipses et recommencer le processus de division.

Il faut noter que cette approche permet de conserver la règle utilisée dans les *treemaps* et qui impose que le ratio des surfaces des figures associées aux noeuds de niveau n soit identique

au ratio des poids associés à ces nœuds. En effet, comme la surface d'une ellipse se calcule avec la formule : $S = \pi a b$ (où a et b sont respectivement les demi-grand axe et demi-petit axe de l'ellipse), l'algorithme de division présenté ci-dessus conserve la règle des ratios.

Cependant, à ce stade, apparaît un problème esthétique dû au fait que les deux ellipses inscrites dans les rectangles de niveau 1 sont tangentes.

Afin d'éviter ce problème, nous appliquons aux deux rectangles R_1 et R_2 une réduction d'un facteur f selon l'axe de division du rectangle (cf. Fig. 108). Les deux rectangles ainsi créés seront nommés R_1^* et R_2^* .

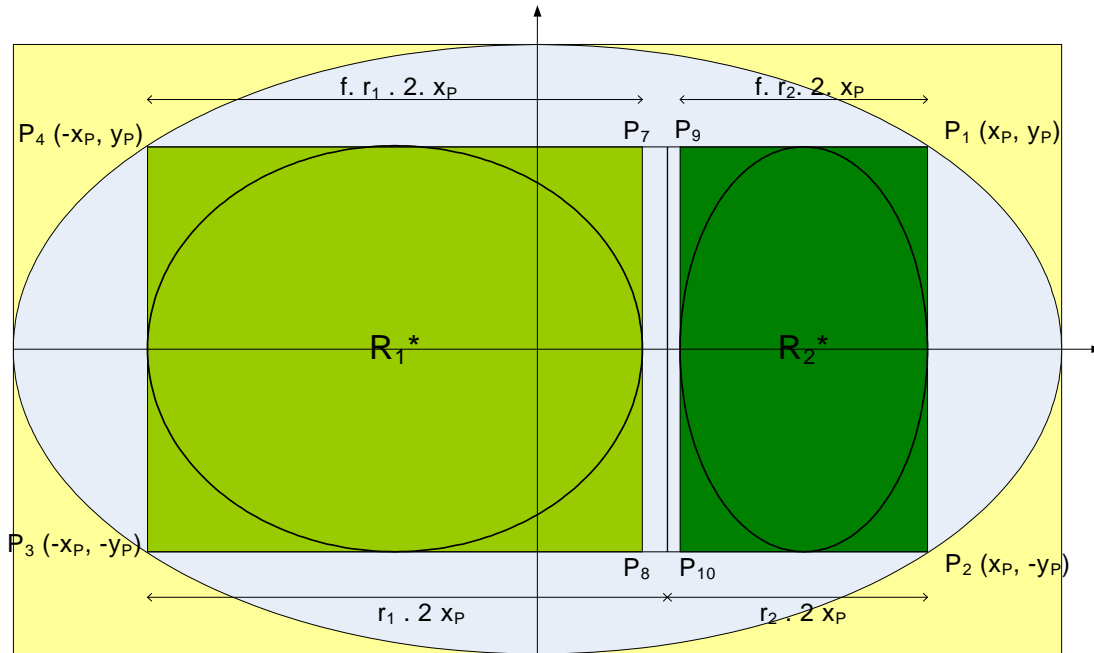


Fig. 108: Suppression du problème de tangence entre les ellipses

Quel que soit le nombre n de rectangles à créer par division du rectangle du niveau supérieur, une distance égale à $2 \cdot x_P \cdot (1-f)$ sera à distribuer de manière égale entre les rectangles du niveau courant pour les espacer. L'espace entre chacun d'entre eux sera donc égal à $2 \cdot x_P \cdot (1-f) / (n-1)$.

Par conséquent, les coordonnées des sommets des n rectangles R_i^* se calculeront avec l'expression :

Sommets du rectangle R_i^* : $\{ S_1^*(x_{i1}^*, y_P), S_2^*(x_{i2}^*, y_P), S_3^*(x_{i2}^*, -y_P), S_4^*(x_{i1}^*, -y_P) \}$ où

$$x_{i1}^* = -x_P + \left[\sum_{j=1}^{(i-1)} (f \cdot r_j \cdot 2 \cdot x_P) \right] + (i-1) 2 \cdot x_P \cdot [(1-f)/(n-1)]$$

$$x_{i2}^* = -x_P + \left[\sum_{j=1}^i (f \cdot r_j \cdot 2 \cdot x_P) \right] + (i-1) 2 \cdot x_P \cdot [(1-f)/(n-1)]$$

Dans notre exemple ($n = 2$), ils seront espacés l'un de l'autre d'une distance de $2 \cdot x_P \cdot (1-f)$.

Les coordonnées de leurs sommets seront donc :

$$R_1^* : \{ P_4 (-x_P, y_P), P_7 (-x_P + f \cdot \frac{4}{3} x_P, y_P), P_8 (-x_P + f \cdot \frac{4}{3} x_P, -y_P), P_3 (-x_P, -y_P) \}$$

$$R_2^* : \{ P_9 (-x_P + f \cdot \frac{4}{3} x_P + 2 x_P (1-f), y_P), P_1 (x_P, y_P), P_2 (x_P, -y_P), P_{10} (-x_P + f \cdot \frac{4}{3} x_P + 2 x_P (1-f), -y_P) \}$$

Grâce à cette simple opération, les ellipses inscrites à ce niveau ne se touchent plus mais elles exploitent une grande partie de l'espace disponible. A titre d'exemple, nous avons appliqué dans la Fig. 108 un facteur $f = 0,95$.

3.2 Choix de l'axe de division inter-niveaux

A ce stade, nous disposons des expressions permettant d'inscrire un rectangle dans une ellipse et vice-versa. Cependant, nous devons encore discuter un élément important : le passage d'un niveau à l'autre ou en d'autres termes, la détermination de l'axe de division du rectangle inscrit dans l'ellipse. Cette problématique fait l'objet de cette section.

a. *Division alternée*

Un premier algorithme de construction des ellimaps se base sur le principe de construction des *treemaps* classiques, c.-à-d. que l'axe de division des rectangles est alterné à chaque niveau. En d'autres termes, si l'axe horizontal a été choisi pour diviser le rectangle inscrit de niveau n , alors l'axe vertical sera choisi comme axe de division au niveau $(n-1)$. Nous appellerons cet algorithme : '*division alternée*'.

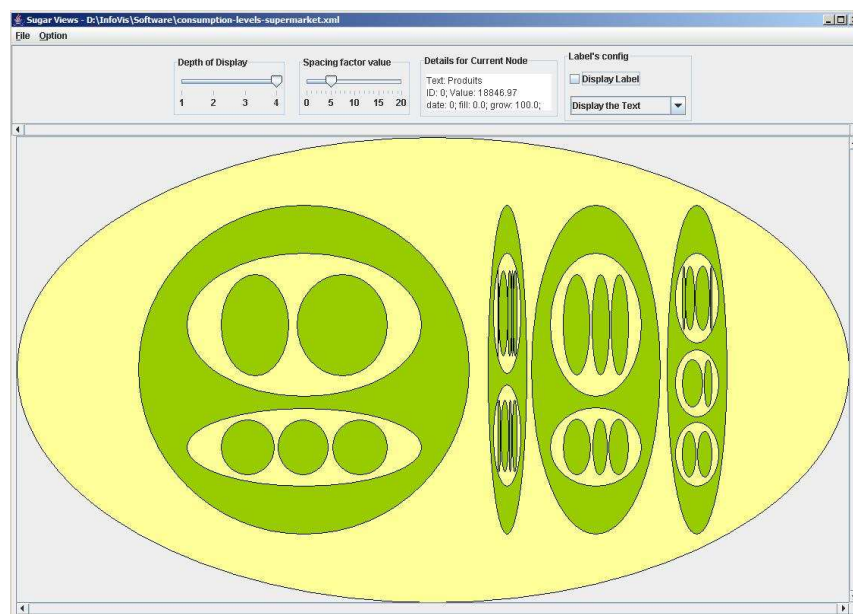


Fig. 109: Ellimap avec algorithme de division alternée

La Fig. 109 illustre ce principe en choisissant l'axe horizontal comme axe pour la division au premier niveau.

L'inconvénient majeur de cette approche est qu'elle ne tient pas compte des dimensions réelles du rectangle à diviser. Une solution sous-optimale apparaît quand la règle impose de diviser le rectangle selon sa plus petite dimension. Dans ce cas, en effet, le ratio longueur – largeur (*aspect ratio*) sera détérioré et les rectangles résultants s'éloigneront de la forme carrée, souvent jugée comme idéale.

b. Division imposée

La seconde approche impose l'axe de division de manière fixe, indépendamment du niveau hiérarchique auquel on se trouve. Il existe deux variantes : la division imposée verticale et la division imposée horizontale. Comme on le remarque sur les Fig. 110 et Fig. 111 cette stratégie est très pénalisante si on considère le ratio d'aspect (*aspect ratio*). Elle n'est à considérer que si des contraintes externes imposent une telle approche.

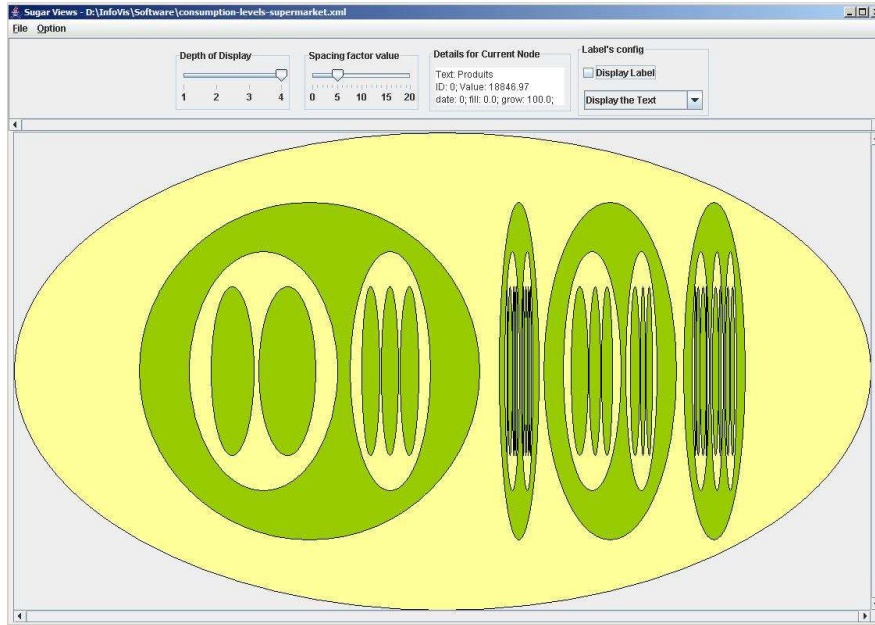


Fig. 110: Ellimap avec algorithme de division imposée horizontale

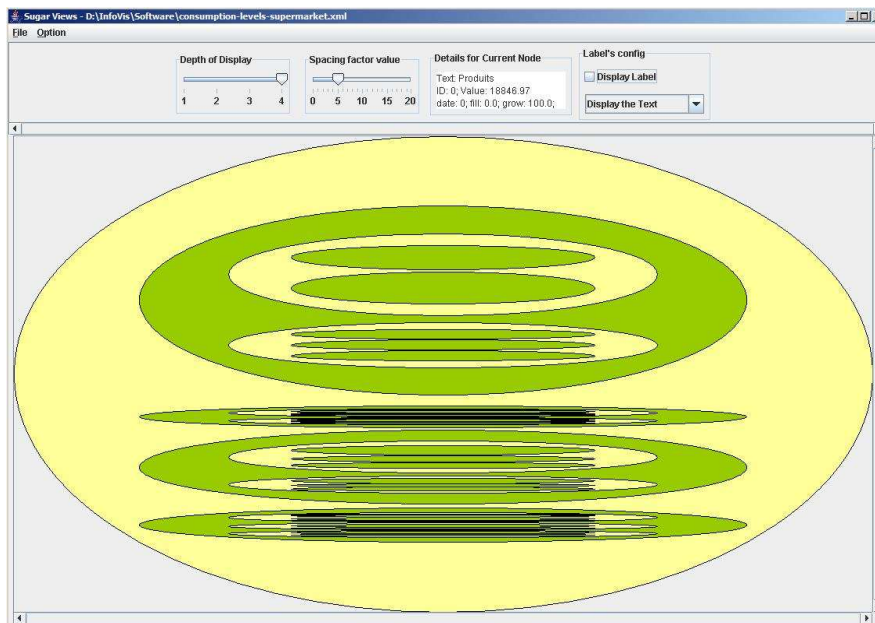


Fig. 111: Ellimap avec algorithme de division imposée verticale

c. Division dépendante

Le troisième algorithme, que nous appellerons ‘division dépendante’, choisit comme axe de division du niveau n celui qui correspond au grand axe de l’ellipse de ce niveau (cf. Fig. 112). En d’autres termes, le rectangle inscrit dans l’ellipse de niveau n est toujours divisé selon sa dimension (horizontale ou verticale) la plus grande.

Cette variante apparaît souvent comme la plus optimale si aucune contrainte particulière ne s’impose à la visualisation car elle tient compte de l’aspect du rectangle à diviser.

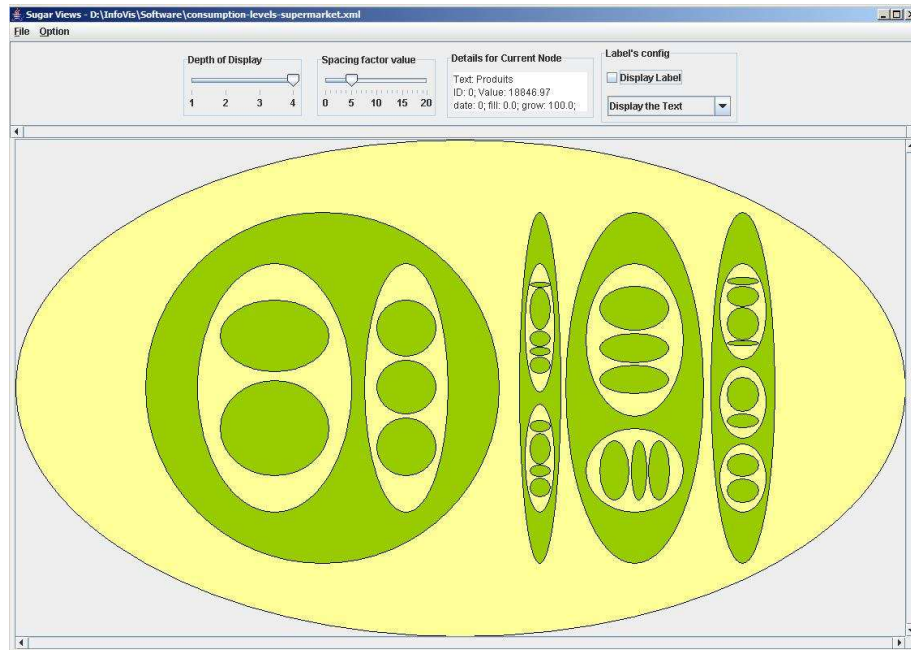


Fig. 112: Ellimap avec algorithme de division dépendante

4. Ellimaps circularisées

Les algorithmes décrits précédemment pour construire les ellimaps dérivent des premiers travaux relatifs aux *treemaps*. Ils en ont hérité des faiblesses.

Depuis leur création au début des années 1990, les *treemaps* ont fait l’objet de diverses améliorations. Parmi celles-ci, la variante ‘*squarified treemap*’²⁸ (Zizi et Beaudoin-Lafon, 1994 ; Bruls et al., 2000) présente un intérêt particulier car elle tend à produire des rectangles imbriqués ayant un rapport longueur-largeur proche de celui de carrés. Selon Bruls et al., cette propriété présente plusieurs avantages.

- ⇒ Le nombre de pixels utilisés pour les bords des rectangles est proportionnel au périmètre et ce nombre est minimal pour un carré.
- ⇒ Les éléments carrés sont plus faciles à détecter et à pointer.
- ⇒ La comparaison de taille de rectangles est plus facile quand leur aspect est similaire.
- ⇒ La précision de la représentation est améliorée.

On évite ainsi la génération de rectangles très fins, ressemblant à des lignes. Nous avons décidé de adapter cette approche au cas des ellimaps car nous rencontrons des problèmes similaires : certaines ellipses peuvent devenir très étroites dans certaines configurations de la

²⁸ L’algorithme ‘*squarified treemap*’ a été initialement proposé par M. Zizi et M. Beaudoin-Lafon mais il a été nommé ainsi et popularisé par Bruls, Huizing et van Wijk.

structure de données. Essayer de rendre les ellipses proches de cercles répond à la même préoccupation que pour des rectangles et conduit aux mêmes avantages.

Afin de faciliter sa compréhension, Bruls et al. ont illustré l'algorithme 'squarified treemap' à partir d'un exemple.

Soit un rectangle de largeur 6 et de hauteur 4. Ce rectangle doit être subdivisé en 7 rectangles de surfaces égales à 6, 6, 4, 3, 2, 2, 1.

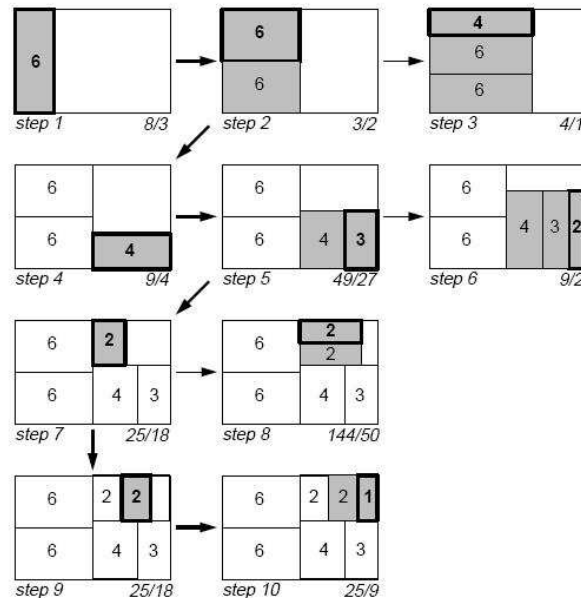


Fig. 113: Algorithme de construction des 'squarified treemaps' (source: [32])

'La première étape consiste à découper le rectangle initial. L'axe de division horizontal est choisi car le rectangle initial est plus large que haut. Ensuite, il faut remplir la moitié gauche du rectangle. Premièrement, on ajoute un premier rectangle dont le rapport d'aspect est de 8/3. Puis, on ajoute un second rectangle, au-dessus du premier. Le rapport d'aspect s'améliore et vaut 3/2. Cependant, si on ajoute le rectangle suivant au-dessus des deux premiers, le rapport d'aspect passe à 4/1. Nous considérons donc que nous avons atteint un optimum pour la partie gauche et nous traitons alors la moitié droite. L'axe de division choisi est l'axe vertical car le rectangle est plus haut que large. A l'étape 4, nous ajoutons le rectangle de surface égale à 4, suivi du rectangle d'aire égale à 3 à l'étape 5. Le rapport d'aspect diminue. Ajouter le rectangle suivant n'améliore pas le résultat, donc nous acceptons le résultat de l'étape 5 et nous recommençons à remplir l'espace en haut à droite. Ces étapes sont répétées jusqu'à ce que tous les rectangles aient été traités.'

Bruls et al. (2000) admettent qu'un résultat optimal ne peut pas être toujours atteint et que des contre-exemples existent. Selon eux, l'ordre dans lequel les rectangles sont traités est important. Ils ont trouvé que trier les rectangles en ordre décroissant produit les meilleurs résultats.

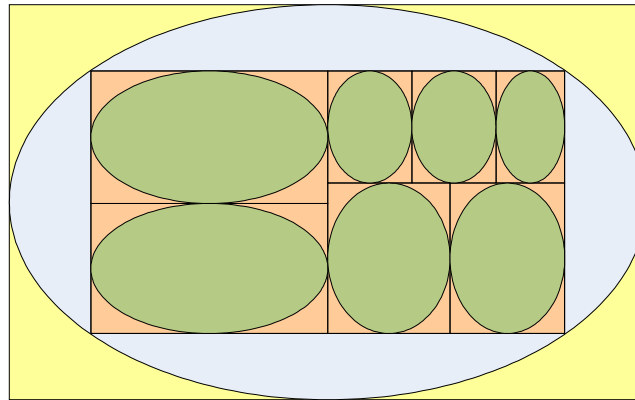


Fig. 114: Ellimaps circularisée (principe)

Nous avons repris l'algorithme '*squarified treemap*' pour le partitionnement des rectangles inscrits dans les ellipses (cf. Fig. 114). Très logiquement, nous avons appelé cet algorithme 'ellimaps circularisées' pour faire référence au fait que nous tendons à rendre les ellipses proches de cercles.

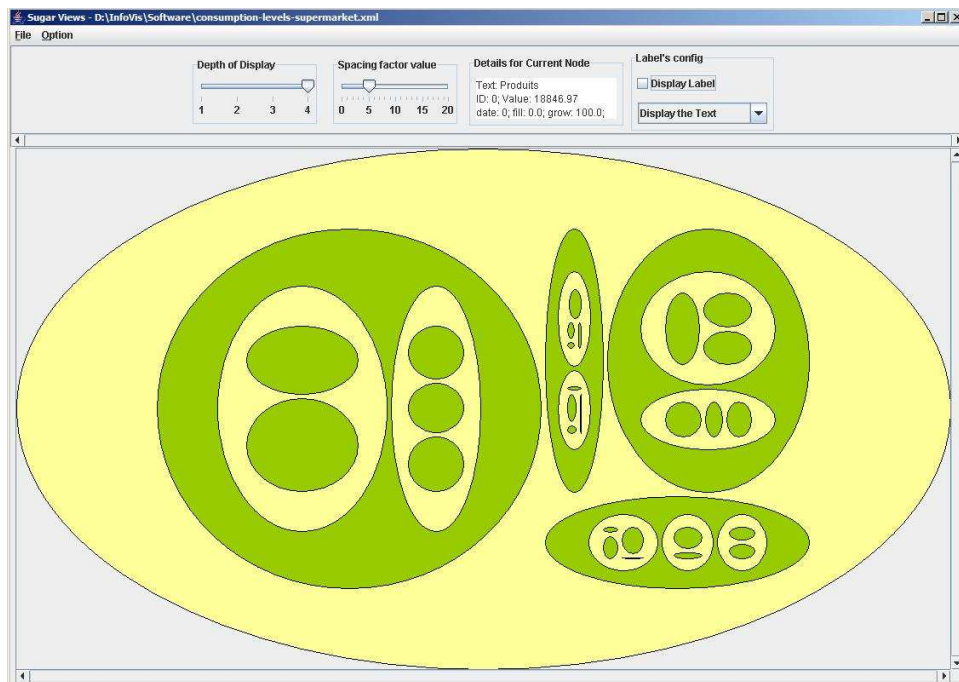


Fig. 115: Ellimaps circularisée

5. Optimisation de l'algorithme

5.1 Lacunes observées

5.1.1. Utilisation de l'espace d'affichage

Les algorithmes de placement des ellipses décrits précédemment présentent des lacunes concernant l'occupation de l'espace disponible. En effet, par rapport aux *treemaps* qui remplissent complètement cet espace, les ellimaps n'en utilisent qu'une fraction. Nous avons donc étudié cette question spécifique en vue de quantifier l'espace non utilisé dans les ellimaps.

La problématique se réduit à deux sous-problèmes fondamentaux :

- quantifier le rapport des surfaces d'un rectangle R et de l'ellipse E qui lui est inscrite,
- quantifier le rapport des surfaces d'une ellipse E et du rectangle inscrit R* dont la longueur et la largeur sont dans le rapport ϕ .

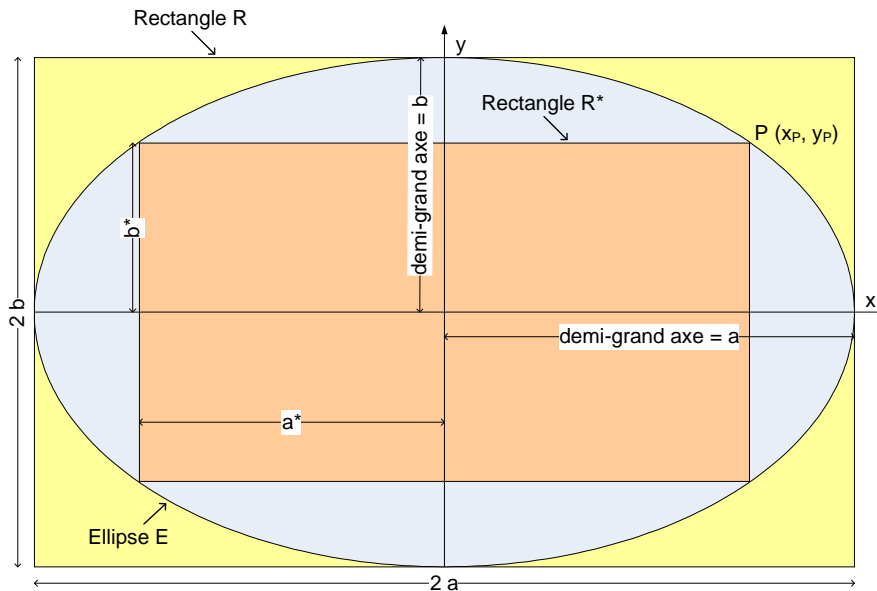


Fig. 116: Rapport des aires entre ellipses et rectangles inscrits (principe)

Si nous conservons les notations utilisées dans les sections précédentes, le premier sous-problème est relatif à l'inscription d'une ellipse E dans un rectangle R de longueur $2a$ et de largeur $2b$. La longueur du demi-grand axe de l'ellipse E vaut a et celle du demi-petit axe vaut b . La surface S_R du rectangle R est naturellement de $4ab$. Par ailleurs, la surface S_E de l'ellipse E a pour valeur πab .

Le ratio des surfaces S_E / S_R se calcule aisément et on obtient la valeur :

$$S_E / S_R = \pi/4 \approx 0,7854$$

En d'autres termes, l'ellipse E n'occupe qu'environ 79 % de l'aire du rectangle R. L'espace non utilisé représente donc environ 21 % de la surface disponible.

Le second sous-problème est lui aussi assez simple. Il nous suffit de calculer la surface S_{R^*} du rectangle R* inscrit dans l'ellipse E dont nous connaissons déjà l'aire. Nous avons calculé

précédemment les valeurs de x_P et y_P qui nous permettent de déterminer les coordonnées des sommets de R^* :

$$x_P = \phi \cdot a \cdot b / \sqrt{(b^2 \cdot \phi^2 + a^2)} \quad \text{et} \quad y_P = a \cdot b / \sqrt{(b^2 \cdot \phi^2 + a^2)}$$

La surface S_{R^*} du rectangle R^* s'élève à $4 x_P y_P$. Si nous développons cette expression, nous obtenons donc la valeur de S_{R^*} .

$$S_{R^*} = 4 x_P y_P = 4 \phi a^2 b^2 / (b^2 \phi^2 + a^2)$$

Le ratio des surfaces S_{R^*} et S_E vaut donc: $4 \phi a b / \pi (b^2 \phi^2 + a^2)$.

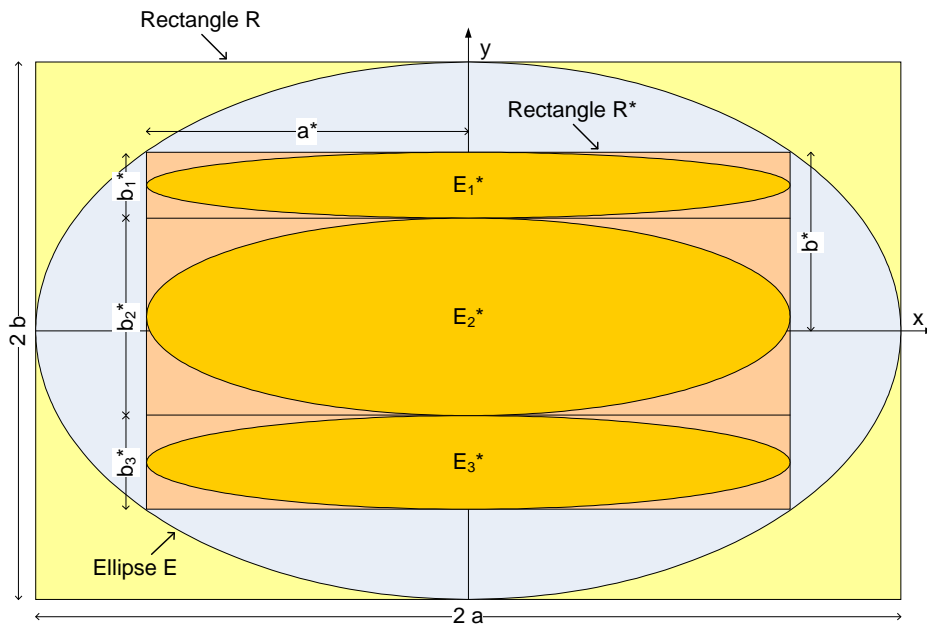


Fig. 117: Rapport des aires entre ellipses et rectangles inscrits (cas général)

Il faut à présent remarquer que l'algorithme de construction des ellimaps présenté précédemment consiste à inscrire dans la plupart des cas plusieurs ellipses E_i^* dans le rectangle R^* . Les résultats généraux que nous venons d'obtenir nous permettent de considérer ce cas (cf. Fig. 117).

L'aire S_i^* d'une ellipse E_i^* est donné par l'expression : $S_i^* = \pi \cdot a^* \cdot (b_i^*/2)$

L'aire totale S_n^* des n ellipses E_i^* se calcule avec l'expression suivante :

$$S_n^* = \sum_{i=1}^n \pi \cdot a^* \cdot (b_i^*/2) = 0,5 \cdot \pi \cdot a^* \cdot \sum_{i=1}^n b_i^* = \pi \cdot a^* \cdot b^*$$

Comme nous connaissons la valeur de a^* et b^* , cette équation devient :

$$S_n^* = \pi \phi a^2 b^2 / (b^2 \phi^2 + a^2)$$

Ce résultat n'est pas dépendant du nombre d'ellipses E_i^* inscrites dans R^* . Il est également valide pour les ellimaps circularisées. En effet, dans ce cas, le rectangle R^* est aussi divisé en sous-rectangles qui couvrent l'entièreté de sa surface. Même si l'expression du calcul de S_n^* diffère, les ellipses E_i^* seront toujours dans le même rapport de surface avec les rectangles R_i^* dans lesquels elles sont inscrites.

Pour être complet, nous devons intégrer le fait que, pour supprimer le problème de tangence des ellipses, nous avons introduit le facteur f qui restreint l'espace disponible pour inscrire les ellipses de niveau $n+1$ dans le rectangle de niveau n .

$$S_n^* = f \pi \varphi a^2 b^2 / (b^2 \varphi^2 + a^2)$$

Dès lors, le ratio entre les surfaces des ellipses E_i^* et du rectangle R^* s'établit à :

$$S_n^* / S_{R^*} = f \pi / 4 \approx 0,7854 f$$

Si nous combinons ces résultats, nous pouvons conclure que si au niveau n de l'arbre des données, la surface S_E est disponible pour l'affichage des données, la surface S_n^* associée au niveau $n+1$ se calculera avec l'expression :

$$S_n^* / S_E = f \varphi a b / (b^2 \varphi^2 + a^2)$$

Dans le cas particulier où $\varphi = a / b$, cette expression se simplifie et se réduit à :

$$S_n^* / S_E = f / 2$$

Pour les valeurs habituelles du facteur d'espacement (c.-à-d. $f \approx 0,9 - 0,95$) Les ellipses E_i^* ont donc une surface cumulée égale à environ la moitié celle de l'ellipse E . De plus, nous avons montré précédemment que cette valeur de φ est celle qui maximise la surface du rectangle R^* et donc de l'ellipse E^* .

5.1.2. Effet de distorsion inter-niveaux

Outre la sous-utilisation de l'espace dédié à l'affichage des données, la réduction de l'espace disponible à chaque niveau hiérarchique induit un effet de bord assez pénalisant : deux nœuds de même poids seront représentés par des ellipses de taille différente s'ils se situent à des niveaux hiérarchiques différents. Par exemple, sur la Fig. 118, les nœuds ID 3 et ID 5 ont tous deux un poids égal à 2 mais le nœud ID 3 occupe une surface plus grande car il se situe à un niveau hiérarchique plus proche de la racine.

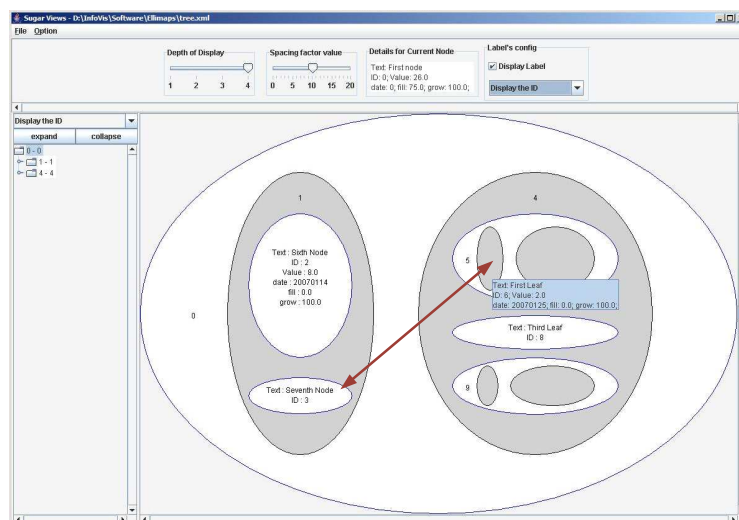


Fig. 118: Problème de la representation non homogène du poids des nœuds (ellimap)

Ce problème est un inconvénient connu des représentations basées sur l'inclusion de formes géométriques. Il se pose d'ailleurs aussi avec les *treemaps* bien que de manière moins évidente. Dans ce cas, il apparaît en effet dès que les contours des rectangles possèdent une certaine épaisseur ou dès qu'une zone inoccupée est laissée entre ces rectangles. Ces facteurs réduisent en effet l'espace disponible pour la surface représentant le poids des nœuds. Les Fig. 119, Fig. 120 et Fig. 121 illustrent ce phénomène. Dans ces trois figures, le même jeu de données est visualisé par un *treemap*. La seule variation concerne l'espace dédié aux contours des rectangles et à l'espace inter-rectangle, c.-à-d. l'espace dévolu à la représentation de la structure des données. Nous avons choisi trois exemples dans lesquels on alloue respectivement 5%, 10 % et 15 % de l'espace d'affichage à la zone inter-rectangle. Dans la terminologie adoptée pour les ellimaps, cela correspond à un facteur d'espacement $f=0,05$; $f=0,10$ et $f=0,15$. Ces valeurs peuvent sembler excessives mais nous visons surtout à ce niveau à illustrer le phénomène à des fins didactiques.

Afin de mieux visualiser le problème, nous avons coloré avec la même couleur les nœuds ayant le même poids :

- rouge \rightarrow poids = 1
- bleu \rightarrow poids = 2
- jaune \rightarrow poids = 3
- vert \rightarrow poids = 4
- ...

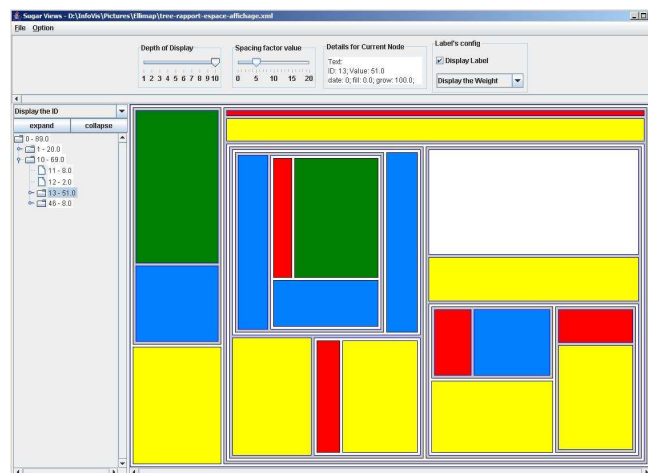


Fig. 119: Problème de la représentation non homogène du poids des nœuds ($f = 0,05$)

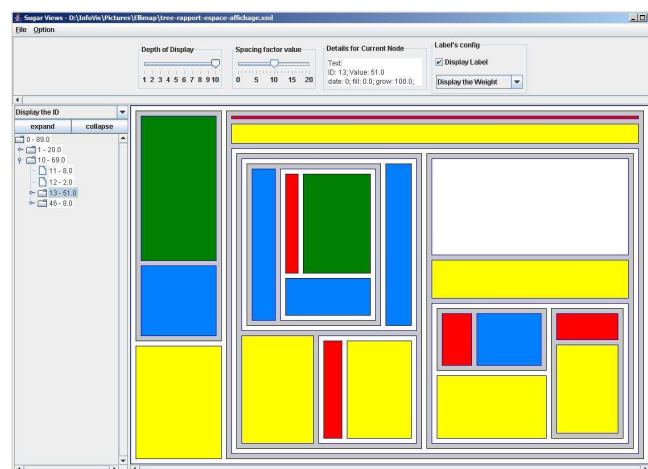


Fig. 120 : Problème de la représentation non homogène du poids des nœuds ($f = 0,10$)

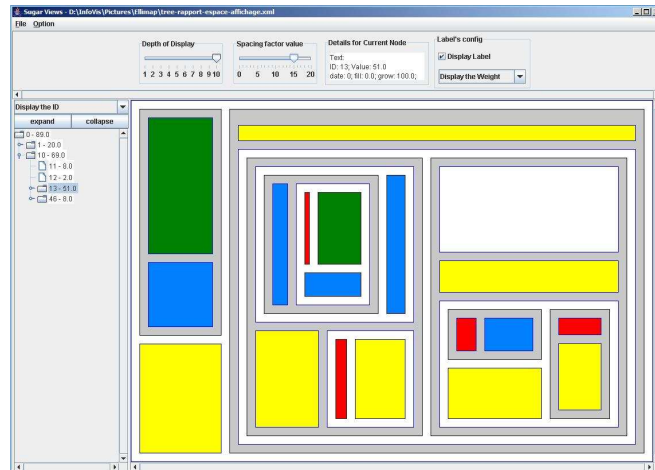


Fig. 121: Problème de la représentation non homogène du poids des nœuds ($f = 0,15$)

La distorsion est d'autant plus grande que l'espace entre les formes géométriques associées aux nœuds (rectangles ou ellipses) est important. Cet espace dépend essentiellement de deux paramètres : le niveau de profondeur du nœud et l'espace laissé libre entre les nœuds à chaque niveau.

L'erreur n'est quasiment pas perceptible dans les treemaps sans bordure (rectangle délimités par un trait fin). Par contre, dans les ellimaps ou dans les *treemaps* avec bordure l'erreur de représentation des poids peut rapidement devenir critique.

Nous pouvons cependant atténuer la portée de cette lacune si on se limite à comparer des nœuds de niveaux hiérarchiques proches ou d'un même niveau. En effet, dans ce cas, la distorsion sera respectivement réduite ou annulée.

En conclusion, si on souhaite rendre plus visible la structure de la hiérarchie en y allouant une fraction de l'espace d'affichage, il faut être conscient que c'est au détriment de la capacité de la représentation à permettre une comparaison aisée des poids de nœuds appartenant à des niveaux hiérarchiques différents.

Par conséquent, nous avons donc légitimement cherché à améliorer nos algorithmes initiaux de construction des ellimaps pour mieux exploiter l'espace d'affichage disponible d'une part et pour minimiser l'effet de distorsion de l'affichage des poids d'autre part.

5.2 Algorithme d'optimisation DGC

Pour améliorer l'utilisation de l'espace d'affichage tout en conservant l'idée originale des ellimaps, nous avons cherché à modifier l'algorithme de base qui détermine les dimensions et le placement des ellipses E_i de niveau $n+1$ dans une ellipse E de niveau n .

Après diverses tentatives, nous avons abandonné la piste de la résolution de systèmes d'équations visant à déterminer la taille et la position des ellipses E_i satisfaisant aux contraintes suivantes :

- les ellipses E_i sont inscrites dans l'ellipse E ;
- les surfaces des ellipses E_i sont dans des rapports imposés ;
- les ellipses E_i occupent la plus grande surface possible tout en répondant aux deux premières conditions.

L'approche par résolution analytique des systèmes d'équations sous-jacents conduisait à des calculs complexes et nous ne sommes même pas certains de l'existence d'une solution analytique. Par ailleurs, notre tentative de résolution numérique des systèmes d'équations ne nous a pas permis d'aboutir à une solution efficace et viable.

La résolution de ce problème par la programmation logique (Van Hentenryck, 1989) a également été abordée bien qu'assez rapidement. Sous réserve d'une analyse plus approfondie, il est apparu que le problème combine des contraintes linéaires et non linéaires et que sa résolution par programmation logique devient elle aussi rapidement très complexe. Néanmoins, cette approche constitue une piste digne d'être étudiée en détail dans des travaux de recherche ultérieurs.

Dès lors, nous avons assoupli la troisième contrainte. Nous n'exigeons plus que les ellipses E_i occupent la plus grande surface possible mais seulement qu'elles occupent une surface cumulée supérieure à celle occupée dans les algorithmes précédemment décrits.

Nous décrivons dans la section suivante l'approche que nous avons élaborée pour répondre à cette question.

D'un point de vue général, ce nouvel algorithme se base sur des déformations géométriques combinées (d'où vient son acronyme : DGC) qui conservent le ratio des surfaces des ellipses E_i . De plus, il évite les difficultés rencontrées pour initialiser le processus d'inscription des ellipses E_i en adoptant comme situation initiale celle résultant de l'algorithme simple d'inscription dans un rectangle avec division dépendante (c.-à-d. selon la dimension la plus grande, cf. Fig. 112). Cette situation de départ est illustrée par la Fig. 122.

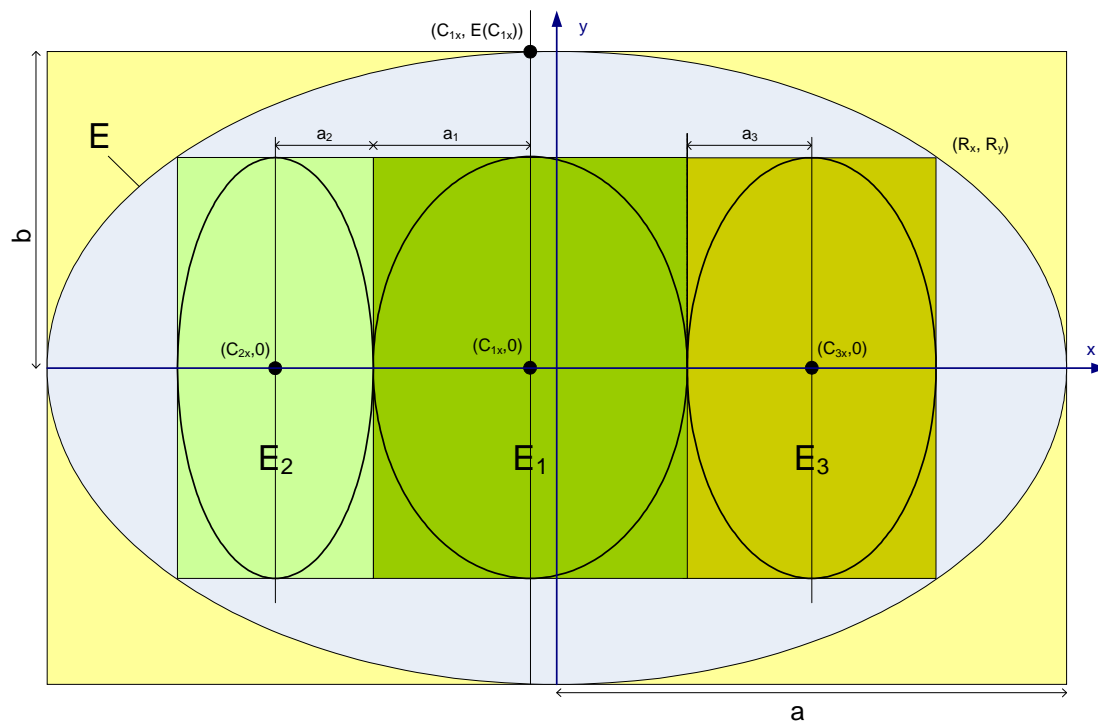


Fig. 122: Algorithme DGC de placement optimisé des ellipses: étape initiale

L'ellipse E est définie par l'équation : $(x^2 / a^2) + (y^2 / b^2) = 1$

Par hypothèse, dans tous les développements relatifs à l'algorithme DGC, l'axe de division du rectangle inscrit est l'axe horizontal (coordonnées en x). En pratique, si pour une ellipse donnée, l'axe de division est l'axe vertical (coordonnées en y), un simple changement de repère permet d'adapter les expressions mathématiques.

La première opération vise à utiliser une proportion plus importante de l'espace libre situé au-dessus et en-dessous du rectangle inscrit. Dans ce but, nous déformons l'ellipse la plus proche du centre de l'ellipse E en imposant que l'aire de cette ellipse reste constante.

L'ellipse à déformer (c.-à-d. E_1 dans notre exemple) est identifiée comme celle dont la distance selon l'axe de division du rectangle (soit l'axe des x dans notre exemple) entre son

centre $C_i(C_{ix}, 0)$ et l'origine des axes $(0,0)$ est minimum. Dans notre exemple, cette condition s'exprime²⁹ : E_j telle que $C_{jx} = \min(|C_{ix}|)$.

La déformation de E_1 donne naissance à l'ellipse E_{1d} (cf. Fig. 123).

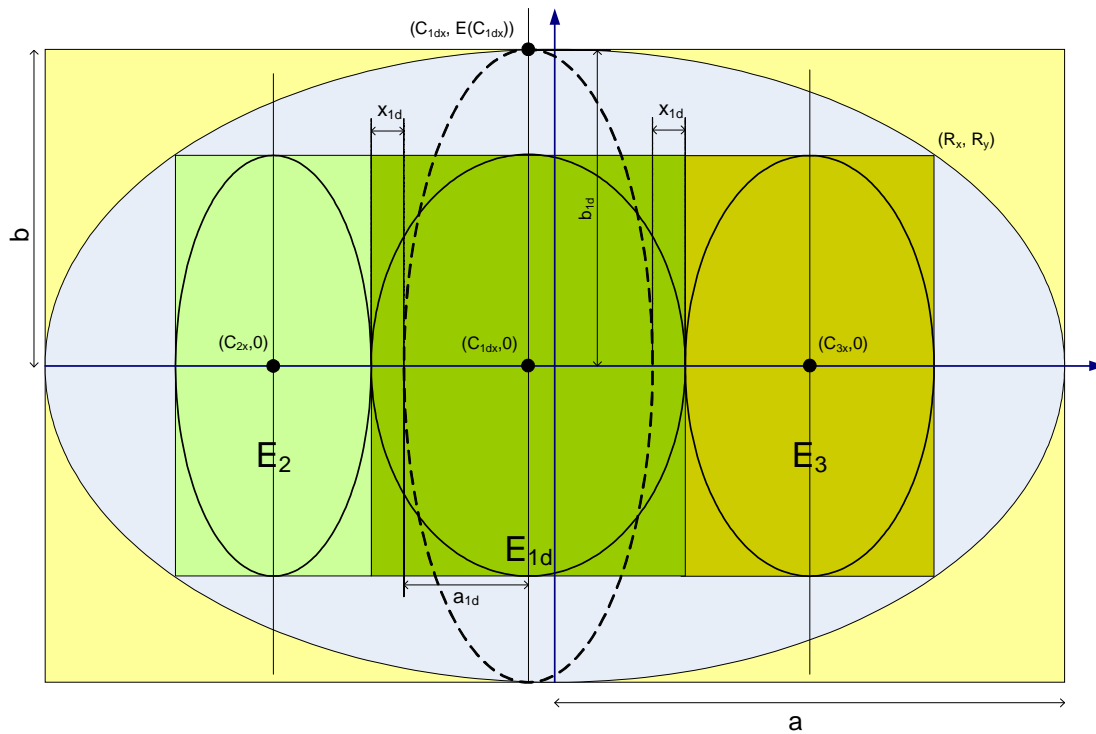


Fig. 123: Algorithme DGC: 1^{ère} étape

L'ellipse E_{1d} est centrée au même point que l'ellipse E_1 , c.-à-d. au point $(C_{1x}, 0)$. Pour faciliter la compréhension, nous renommeons ce point $(C_{1dx}, 0)$.

Pour former l'ellipse E_{1d} , l'ellipse E_1 est déformée selon l'axe des y de manière à ce qu'elle devienne tangente à l'ellipse E au point $(C_{1dx}, E(C_{1dx}))$.

Comme nous connaissons l'équation de définition de E , nous pouvons calculer $E(C_{1dx})$ avec l'expression : $E(C_{1dx}) = (b/a) \sqrt{a^2 - C_{1dx}^2}$.

Notons que, comme dans la plupart des cas, $C_{1dx} \neq 0 \Rightarrow E(C_{1dx}) \neq b$.

A présent que $E(C_{1dx})$ est connu, nous pouvons calculer la valeur du facteur de déformation D_{1y} appliquée à E_1 selon l'axe des y : $D_{1y} = E(C_{1dx}) / R_y$

Nous connaissons aussi (cf. Fig. 106) la valeur de $R_y = a \cdot b / \sqrt{b^2 \cdot \phi^2 + a^2}$

Comme nous avons adopté la valeur $\phi = a/b$, nous obtenons $R_y = b / \sqrt{2}$

Dès lors, nous concluons que $D_{1y} = (\sqrt{2}/a) \cdot \sqrt{a^2 - C_{1dx}^2}$

Comme les aires des ellipses E_1 et E_{1d} doivent être égales, il faut aussi appliquer à E_1 un facteur de déformation D_{1x} selon l'axe des x. Ce facteur doit compenser D_{1y} et donc $D_{1x} = 1 / D_{1y}$.

En conclusion, l'ellipse E_{1d}

- est centrée en $(C_{1dx}, 0)$
- possède un demi-grand axe b_{1d} égal à $(b/a) \cdot \sqrt{a^2 - C_{1dx}^2}$
- et possède un demi-petit axe a_{1d} égal à $(D_{1x} \cdot a_1)$ soit $a \cdot a_1 / [\sqrt{2} \cdot \sqrt{a^2 - C_{1dx}^2}]$

²⁹ $|a|$ est la valeur absolue de a .

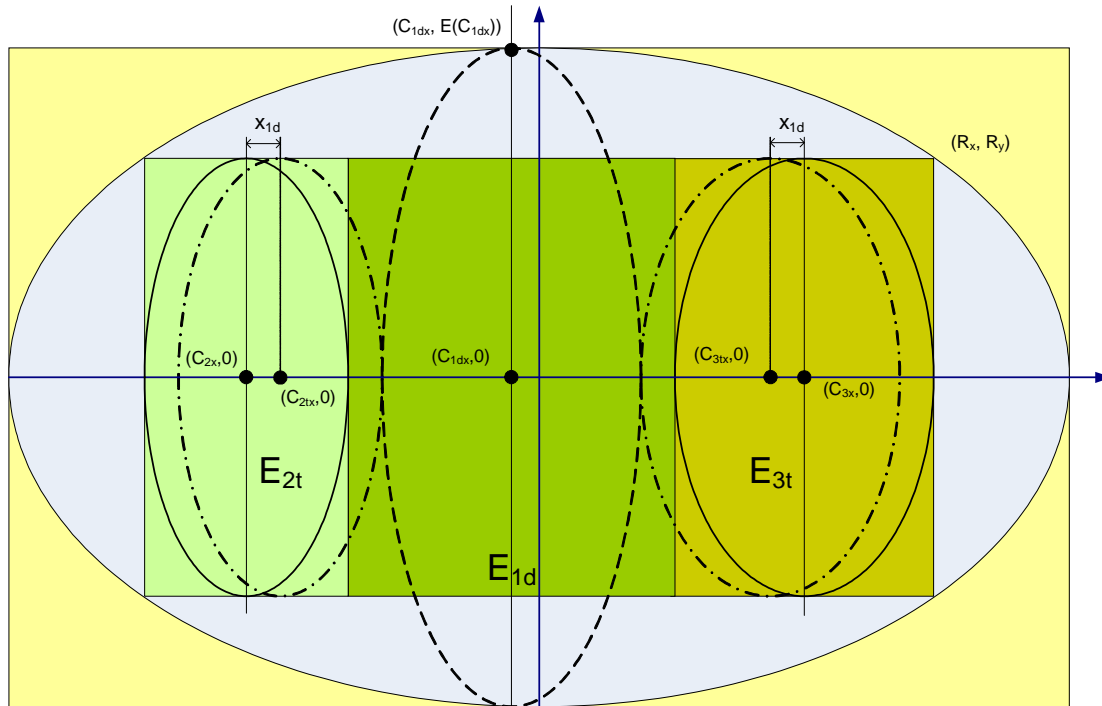


Fig. 124: Algorithme DGC: 2^{ème} étape

La seconde étape consiste à tradater toutes les ellipses E_i (à l'exception de E_1 , devenue E_{1d}) pour les rendre tangentes les unes aux autres le long de l'axe des x . Dans notre exemple (cf. Fig. 124), cela revient à tradater E_2 et E_3 d'une distance égale à X_{1d} . Nous nommerons les deux nouvelles ellipses ainsi créées E_{2t} et E_{3t} .

La valeur X_{1d} est connue : $X_{1d} = a_1 \cdot (1 - D_{1x})$

Nous pouvons en déduire les positions de E_{2t} et E_{3t} :

E_{2t} est une ellipse centrée en $(C_{2x} + X_{1d}, 0)$, de demi-grand axe R_y et de demi-petit axe a_2

E_{3t} est une ellipse centrée en $(C_{3x} - X_{1d}, 0)$, de demi-grand axe R_y et de demi-petit axe a_3

En adoptant le formalisme $C_{2tx} = C_{2x} + X_{1d}$ et $C_{3tx} = C_{3x} - X_{1d}$, nous obtenons :

E_{2t} est une ellipse centrée en $(C_{2tx}, 0)$, de demi-grand axe R_y et de demi-petit axe a_2

E_{3t} est une ellipse centrée en $(C_{3tx}, 0)$, de demi-grand axe R_y et de demi-petit axe a_3

La 3^{ème} étape a pour objectif de déformer les ellipses E_{2t} et E_{3t} de manière similaire à la déformation appliquée à E_1 lors de la première étape. Nous détaillerons le processus pour le cas de l'ellipse E_{2t} . Le processus est semblable pour les autres ellipses E_{it} .

L'ellipse résultant de la déformation de E_{2t} sera appelée E_{2td} .

L'ellipse E_{2td} est centrée au même point que l'ellipse E_{2t} , c.-à-d. au point $(C_{2tx}, 0)$. Pour faciliter la compréhension, nous renommons ce point $(C_{2tdx}, 0)$.

Pour former l'ellipse E_{2td} , l'ellipse E_{2t} est déformée selon l'axe des y de manière à ce qu'elle devienne tangente à l'ellipse E au point $(C_{2tdx}, E(C_{2tdx}))$.

Comme nous connaissons l'équation de définition de E , nous pouvons calculer $E(C_{2tdx})$ avec l'expression : $E(C_{2tdx}) = (b/a) \sqrt{a^2 - C_{2tdx}^2}$.

A présent que $E(C_{2tdx})$ est connu, nous pouvons calculer la valeur du facteur de déformation D_{2y} appliquée à E_{2t} selon l'axe des y : $D_{2y} = E(C_{2tdx}) / R_y$

Nous connaissons la valeur de $R_y = b / \sqrt{2}$

Dès lors, nous concluons que $D_{2y} = (\sqrt{2} / a) \cdot \sqrt{(a^2 - C_{2tdx}^2)}$

Comme les aires des ellipses E_{2t} et E_{2td} doivent être égales, il faut aussi appliquer à E_{2t} un facteur de déformation D_{2x} selon l'axe des x. Ce facteur doit compenser D_{2y} et donc $D_{2x} = 1 / D_{2y}$.

En conclusion, l'ellipse E_{2td}

- est centrée en $(C_{2tdx}, 0)$
- possède un demi-grand axe égal à $b_{2d} = (b / a) \cdot \sqrt{(a^2 - C_{2tdx}^2)}$
- et possède un demi-petit axe égal à $a_{2d} = (D_{2x} \cdot a_2) = a \cdot a_2 / [\sqrt{2} \cdot \sqrt{(a^2 - C_{2tdx}^2)}]$

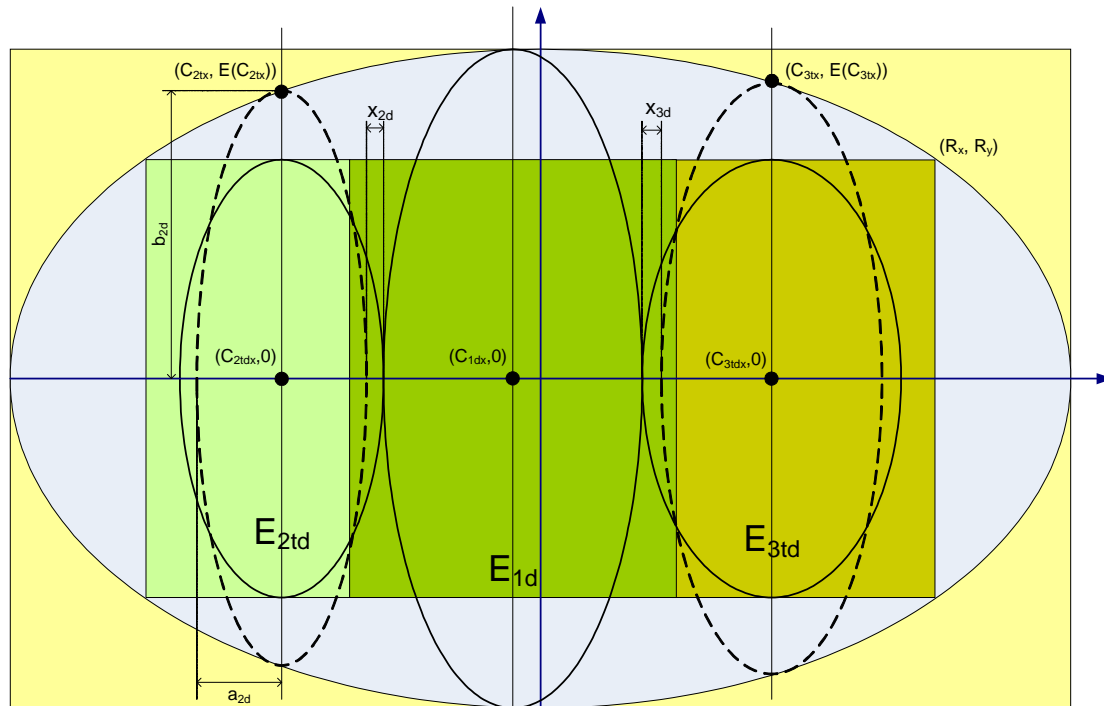


Fig. 125: Algorithme DGC: 3^{ème} étape

La 4^{ème} étape est le pendant de la seconde étape pour les ellipses E_{itd} ($i \neq 1$). Elle consiste à tradater les ellipses E_{itd} (c.-à-d. E_{2td} et E_{3td} dans notre exemple) pour rendre toutes les ellipses tangentes entre elles (cf. Fig. 126).

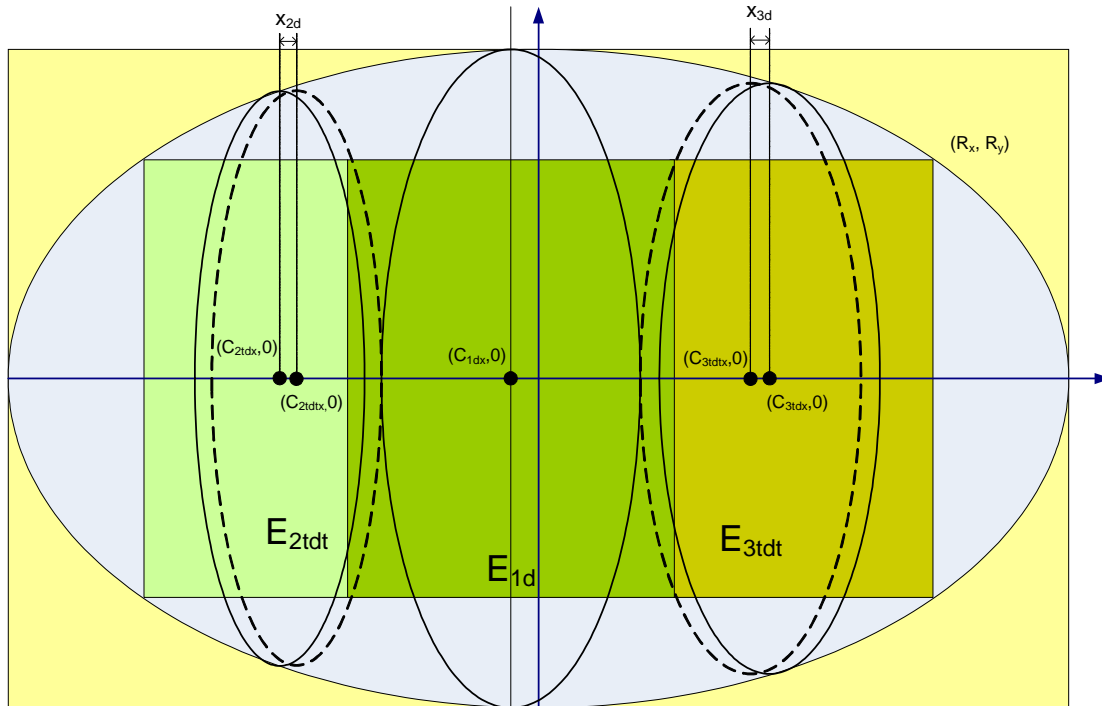
L'ellipse E_{2td} doit être tradatée de la distance x_{2d} pour donner naissance à l'ellipse E_{2tdt} .

Nous connaissons la valeur de x_{2d} : $x_{2d} = a_2 \cdot (1 - D_{2x})$

Le centre de l'ellipse E_{2tdt} est le point $(C_{2tdx} + x_{2d}, 0)$ que nous renommerons $(C_{2tdtx}, 0)$.

Son demi grand axe vaut b_{2d} et son demi-petit axe vaut a_{2d} . Ces deux valeurs ont été calculées à l'étape précédente.

Un développement semblable s'applique à la translation de l'ellipse E_{2td} pour donner l'ellipse E_{3tdt} .


 Fig. 126: Algorithme DGC: 4^{ème} étape

Au cours des étapes 1 à 4, nous avons cherché à occuper une plus grande partie de l'espace libre au-dessus et en-dessous du rectangle. Les étapes suivantes vont réaliser une opération similaire pour les espaces non utilisés à gauche et à droite de celui-ci.

Notons que jusqu'à présent la surface cumulée des ellipses E_i est égale à celle des ellipses inscrites dans le rectangle car les opérations géométriques que nous avons réalisées ont conservé les aires des ellipses constantes. Par contre dans les étapes suivantes, nous modifierons les surfaces des ellipses (sans modifier leurs rapports). C'est donc à partir de ce point du processus DGC que nous engrangerons un gain en termes d'occupation d'espace.

La 5^{ème} étape déforme l'ellipse la plus proche du centre (c.-à-d. E_{1d}) selon l'axe des x . Cette nouvelle ellipse sera appelée E_{1dm} (cf. Fig. 127).

Nous ne déplaçons pas son centre et donc l'ellipse E_{1dm} est centrée au point $(C_{1dx}, 0)$ que nous renommerons $(C_{1dmx}, 0)$.

Comme notre objectif est d'occuper la plus grande proportion possible de l'espace libre, nous déformerons les différentes ellipses de manière à ce que la somme de leurs axes selon l'axe des x soit égale à l'axe de l'ellipse E selon la même direction. Cependant, les ratios de leurs aires devront rester constants.

L'espace disponible x_{disp} vaut : $x_{disp} = 2 \cdot x_{init} + 2 \cdot x_{1d} + 2 \cdot x_{2d} + 2 \cdot x_{3d}$

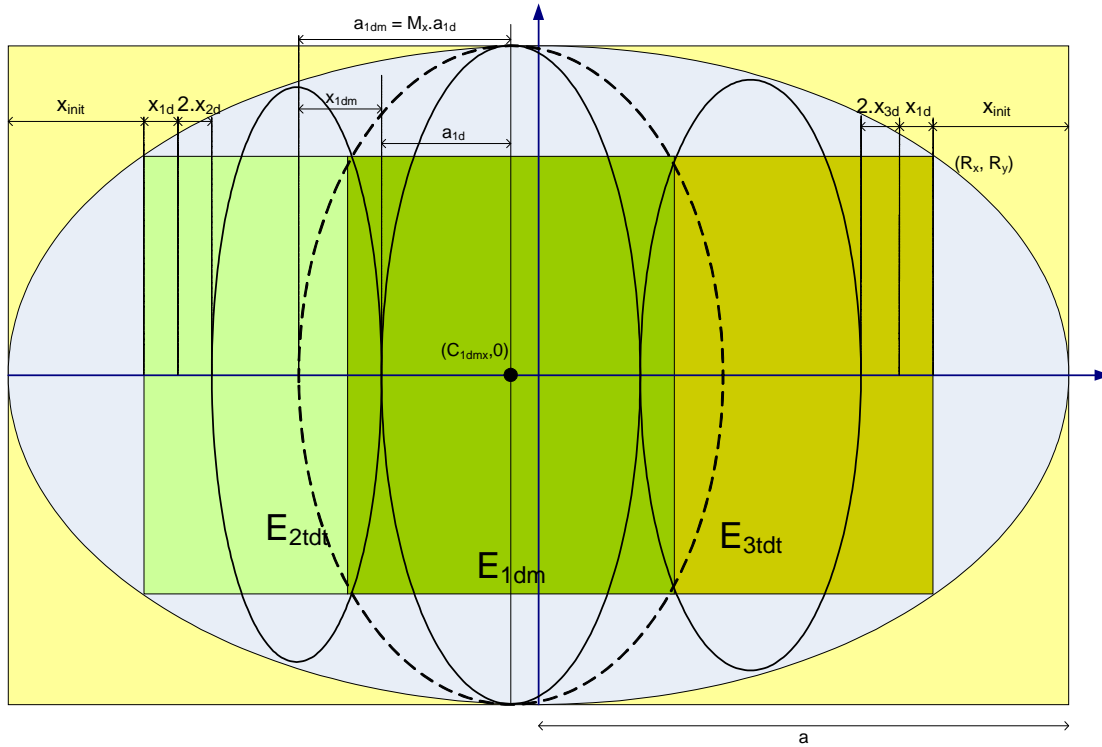
Toutes ces valeurs sont connues, y compris x_{init} : $x_{init} = (a - R_x)$

or nous connaissons (cf. Fig. 106) la valeur³⁰ de $R_x = a / \sqrt{2}$ et donc $x_{init} = a \cdot (1 - 1/\sqrt{2})$

Nous appliquons selon l'axe des x un facteur de dilatation identique M_x à toutes les ellipses afin de conserver constant le ratio de leurs aires.

Ce facteur se calcule avec l'expression : $M_x = 2 \cdot a / (2 \cdot a - x_{disp})$

³⁰ avec la valeur $\phi = a / b$


 Fig. 127: Algorithme DGC: 5^{ème} étape

L'ellipse E_{1dm} est à présent complètement définie :

- elle est centrée en $(C_{1dmx}, 0)$
- la longueur de son demi-grand axe est égale à $b_{1dm} = b_{1d}$
- la longueur de son demi-petit axe est égale à $a_{1dm} = M_x \cdot a_{1d}$

Ensuite, dans la 6^{ème} étape, nous translatons les ellipses E_{2tdt} et E_{3tdt} de la distance x_{1dm} pour les rendre toutes les ellipses tangentes. Nous appellerons les ellipses translatées E_{2tdtt} et E_{3tdtt} . La valeur de x_{1dm} se calcule avec l'expression : $x_{1dm} = a_{1dm} - a_{1d}$.

En conséquence, l'ellipse E_{2tdtt} est centrée en $(C_{2tdt} - x_{1dm}, 0)$ que nous renommons $(C_{2tdtt}, 0)$. Son demi-grand axe a une longueur égale à b_{2d} et son demi-petit axe une longueur égale à a_{2d} .

De manière semblable, l'ellipse E_{3tdtt} est centrée en $(C_{3tdt} + x_{1dm}, 0)$ que nous renommons $(C_{3tdtt}, 0)$. Son demi-grand axe a une longueur égale à b_{3d} et son demi-petit axe une longueur égale à a_{3d} .

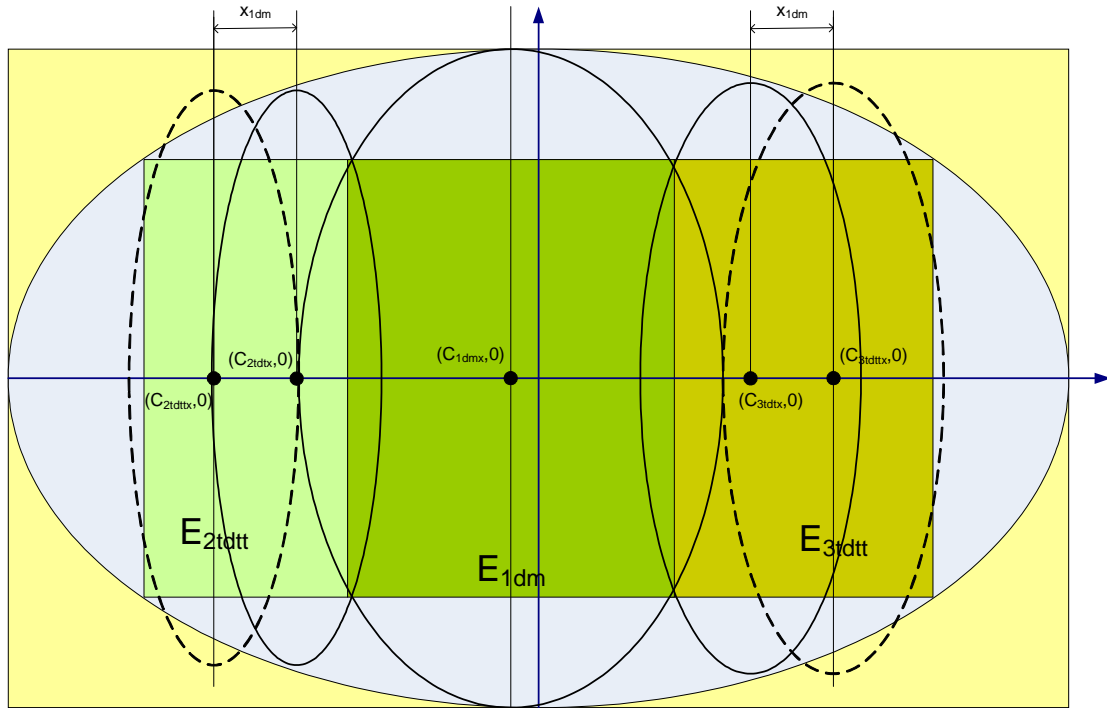


Fig. 128: Algorithme DGC: 6^{ème} étape

La 7^{ème} étape dilate les ellipses E_{itdt} avec le même facteur M_x que celui appliqué à l'ellipse E_{1d} pour donner naissance aux ellipses E_{itdtm} .

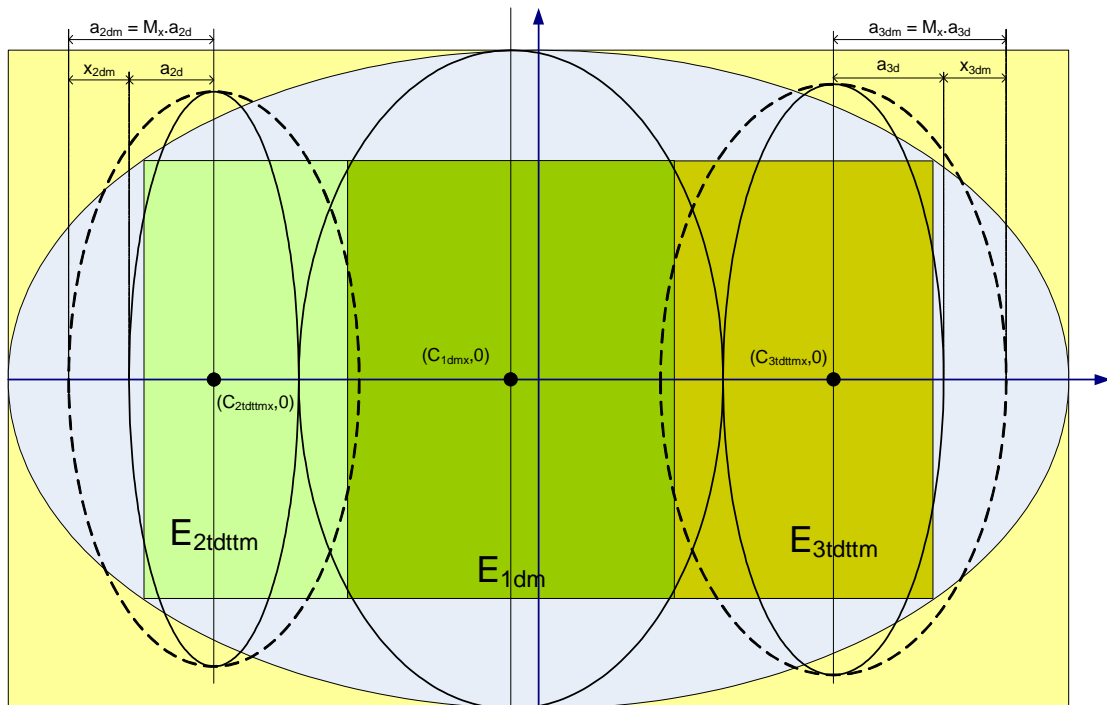


Fig. 129: Algorithme DGC: 7^{ème} étape

L'ellipse E_{2tdtm} prend alors comme valeur pour son demi-petit axe : $a_{2dm} = (M_x \cdot a_{2d})$.
La longueur de son demi-grand axe ne change pas et reste égale à b_{2d} .

Elle reste centrée au même point que $E_{2\text{dtt}}$, soit en $(C_{2\text{dtt}}, 0)$ que nous renommons $(C_{2\text{dttmx}}, 0)$. La même opération s'applique à $E_{3\text{dttm}}$.

La 8^{ème} étape translate à nouveau les ellipses dilatées à l'étape précédente pour rendre toutes les ellipses tangentes.

Pour que l'ellipse $E_{2\text{dttmt}}$ soit tangente à $E_{1\text{dm}}$, il faut traduire $E_{2\text{dttmt}}$ d'une distance égale à $x_{2\text{dm}} = a_{2\text{dm}} - a_{2\text{d}}$.

En résumé, l'ellipse $E_{2\text{dttmt}}$ est donc centrée au point $(C_{2\text{dttmx}} - x_{2\text{dm}}, 0)$ que nous renommons $(C_{2\text{dttmtx}}, 0)$. Son demi-grand axe a une longueur égale à $b_{2\text{d}}$ et son demi-petit axe une longueur égale à $a_{2\text{dm}}$.

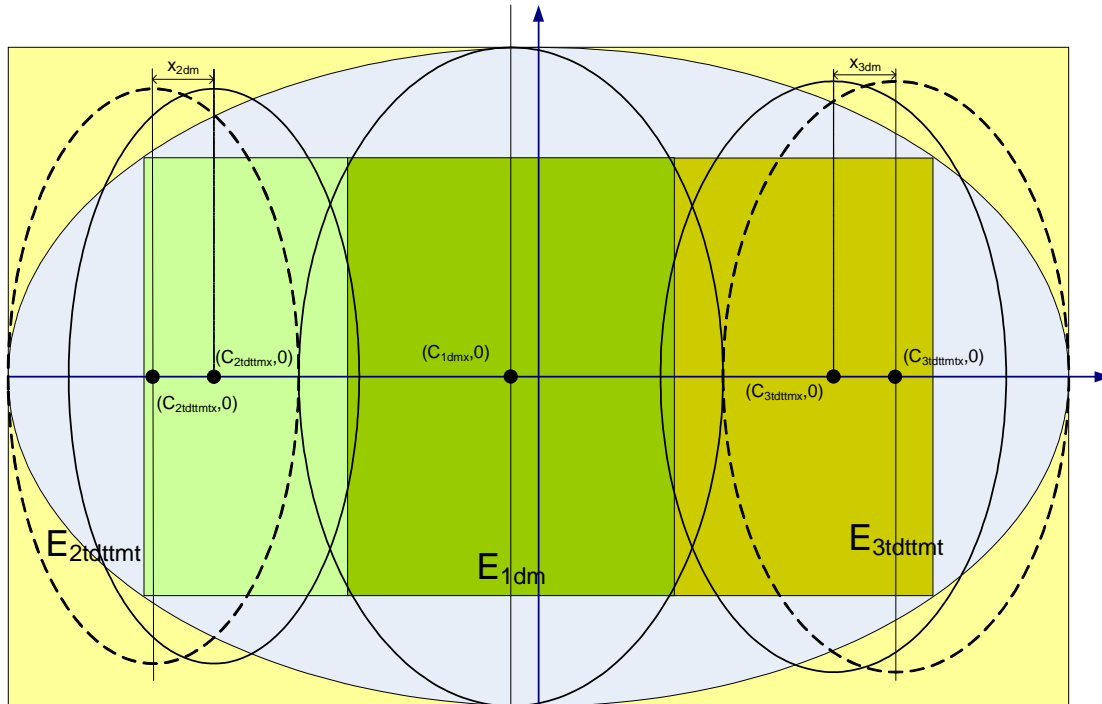


Fig. 130: Algorithme DGC: 8^{ème} étape

Pour que l'ellipse $E_{3\text{dttmt}}$ soit tangente à $E_{1\text{dm}}$, il faut traduire $E_{3\text{dttmt}}$ d'une distance égale à $x_{3\text{dm}} = a_{3\text{dm}} - a_{3\text{d}}$.

Par conséquent, l'ellipse $E_{3\text{dttmt}}$ est centrée au point $(C_{2\text{dttmx}} + x_{3\text{dm}}, 0)$ que nous renommons $(C_{3\text{dttmtx}}, 0)$. Son demi-grand axe a une longueur égale à $b_{3\text{d}}$ et son demi-petit axe une longueur égale à $a_{3\text{dm}}$.

A partir de la 9^{ème} étape, nous adoptons une démarche empirique visant à trouver une solution acceptable pour réduire la taille des ellipses. Mathématiquement, elle ne constitue probablement pas une solution optimale. Cependant, notre expérience nous indique qu'elle est acceptable dans les cas que nous avons testés.

Dans la 9^{ème} étape, nous calculons pour chaque ellipse E_{idttmt} la valeur de y_{ie} (longueur excédentaire selon l'axe des y).

Ces valeurs peuvent être calculées avec l'expression : $y_{\text{ie}} = [E_{\text{idttmt}}(C_{\text{idttmtx}}) - E(C_{\text{idttmtx}})]$

Comme $E_{\text{idttmt}}(C_{\text{idttmtx}}) = b_{2\text{d}}$ et que $E(C_{\text{idttmtx}}) = (b/a) \cdot \sqrt{(a^2 - C_{\text{idttmtx}}^2)}$,

nous connaissons la valeur de $y_{\text{ie}} = [b_{2\text{d}} - (b/a) \cdot \sqrt{(a^2 - C_{\text{idttmtx}}^2)}]$

Ensuite, nous définissons pour chaque ellipse E_{itdtmt} la valeur de y_{ier} (longueur excédentaire relative selon l'axe des y) avec l'expression : $y_{ier} = b_{id} / (b_{id} - y_{ie})$

Nous sélectionnons le maximum des valeurs de y_{ier} , soit $Y_{ir} = \max \{y_{ier}\}$. Si Y_{ir} est inférieur à l'unité, nous passons directement à la 11^{ème} étape³¹.

Par contre dans le cas général où $Y_{ir} > 1$, nous devons réduire la taille des ellipses à inscrire dans E. Cette opération sera effectuée à la 10^{ème} étape.

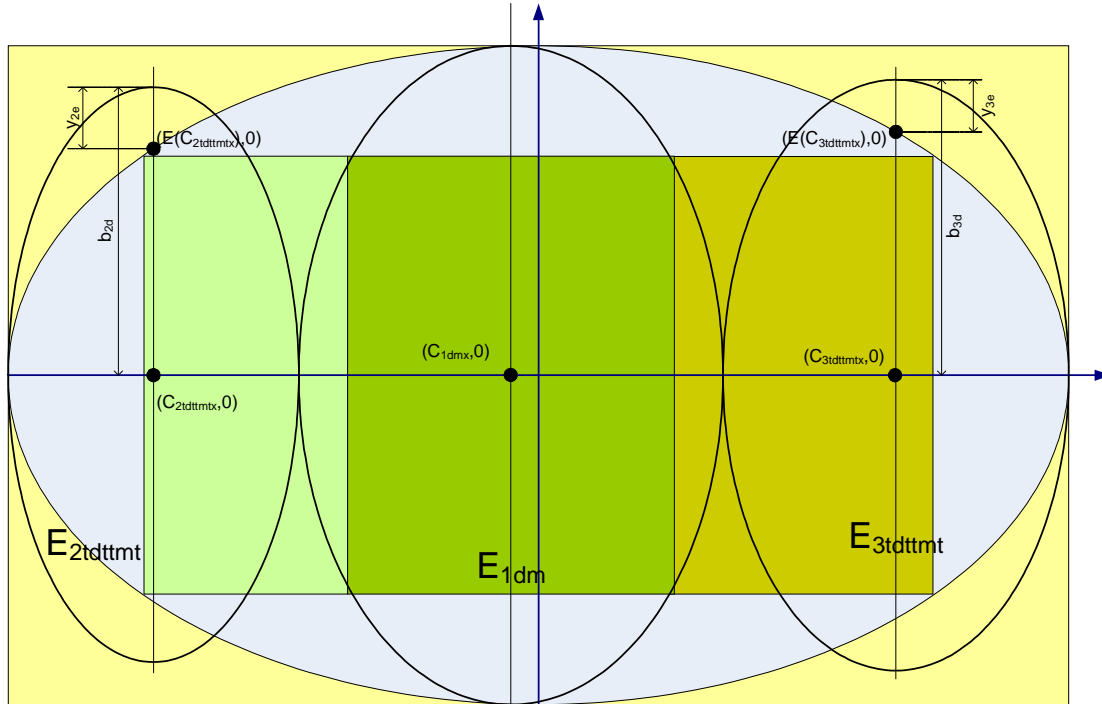


Fig. 131: Algorithme DGC: 9^{ème} étape

A la 10^{ème} étape, nous appliquons un facteur de réduction R_x selon l'axe des x et R_y selon l'axe des y à toutes les ellipses inscrites (y compris E_{1dm}). Le ratio des aires de ces ellipses restera donc constant. Le but des fonctions R_x et R_y est de réduire la taille des ellipses pour qu'elles soient toutes situées à l'intérieur de l'ellipse extérieure E (cf. Fig. 132).

Nous appellerons toutes les ellipses réduites E_{ir} . Une fois réduite E_{1dm} s'appellera E_{1r} et E_{itdtmt} s'appellera E_{ir} .

L'identification des fonctions R_x et R_y est un problème complexe d'optimisation si on souhaite atteindre une maximisation de l'aire cumulée des ellipses E_{ir} .

Nous avons choisi d'aborder ce problème avec une stratégie d'approximation numérique (initialisation à l'étape 10 et itérations à l'étape 11).

Le processus est initialisé à la 10^{ème} étape en calculant un facteur de réduction de la taille des ellipses identique selon les deux axes x et y. Soit $R_x = R_y$ ce facteur de réduction.

Au début de notre réflexion, nous avons pensé qu'il était intéressant d'imposer que R_x soit une fonction du maximum de longueur excédentaire relative Y_{ir} .

³¹ Il faut noter que nous n'affirmons pas que si $Y_{ir} < 1$, toutes les ellipses sont obligatoirement situées à l'intérieur de E car nous pensons (sans l'avoir démontré) qu'il existe des cas critiques pour lesquelles cette affirmation n'est pas valide. Cependant, en pratique, ce critère correspond très souvent à l'inclusion des ellipses dans E.

Nous avons donc expérimenté les fonctions suivantes :

- $R_x = R_y = g(Y_{ir}) = 1 / Y_{ir}$
- $R_x = R_y = g(Y_{ir}) = n / (n - 1 + Y_{ir})$

Cependant, les expériences que nous avons réalisées montrent qu'en général, ces fonctions réduisent trop fortement la taille des ellipses.

Nos essais empiriques nous ont amené à choisir une valeur constante pour les facteurs de réduction R_x et R_y . En pratique, des valeurs comprises entre 0,9 et 0,99 fournissent des résultats acceptables³².

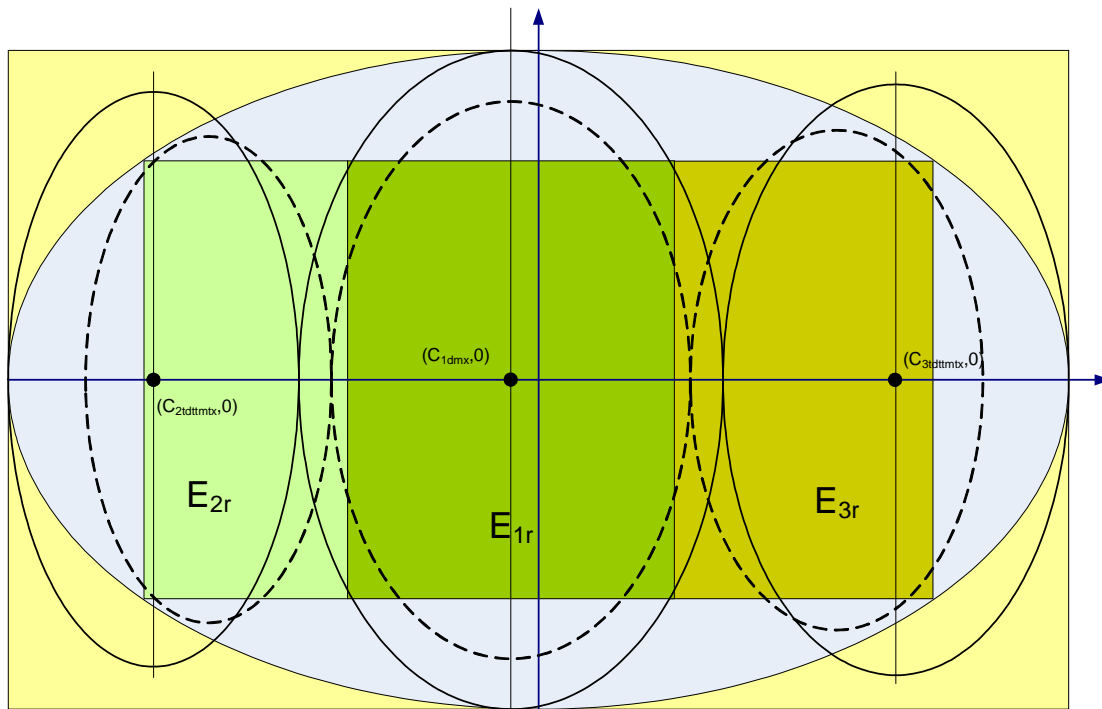


Fig. 132: Algorithme DGC: 10^{ème} étape

Par des calculs similaires à ceux des étapes 1 à 8, la taille et la position de chacune des ellipses E_{ir} peuvent alors être déterminées. En effet, il s'agit toujours de déformer les ellipses selon les deux axes x et y , puis de les translater pour les rendre tangentes selon l'axe des x .

Nous revenons ensuite à la 9^{ème} étape pour évaluer (approximativement³¹) si les ellipses E_{ir} sont toutes situées à l'intérieur de l'ellipse E .

Nous recalculons de nouvelles valeurs y_{ier} pour les longueurs excédentaires relatives selon l'axe y et une nouvelle valeur $Y_{ir} = \max\{y_{ier}\}$.

Si $Y_{ir} < 1$, nous pouvons passer à la 11^{ème} étape.

Par contre, si $Y_{ir} > 1$, nous appliquons de nouveau la 10^{ème} étape.

A la fin de la 10^{ème} étape, le but visé est de disposer d'une situation initiale aussi bonne que possible avant de commencer le processus itératif de la 11^{ème} étape. En effet, comme la 11^{ème} étape est plus intensive en calculs que les précédentes, il est important de minimiser le nombre d'itérations de cette étape. Il ne faut pas perdre de vue que les ellimaps ont pour

³² A titre d'information, notre application prototype était paramétrée avec la valeur $R_x = R_y = 0,98$.

vocation de constituer des représentations graphiques offrant une interactivité en (quasi-) temps réel.

Ensuite, à la 11^{ème} étape, un processus itératif est enclenché. Le principe de base consiste à vérifier si les ellipses E_{ir} ainsi réduites sont situées à l'intérieur de l'ellipse extérieure E . Par rapport aux 9^{ème} et 10^{ème} étapes, nous imposons un respect plus strict de cette condition. Pour chacune des ellipses E_{ir} , nous calculerons si un nombre³³ limité de points de son périmètre sont inclus dans E (cf. Fig. 133).

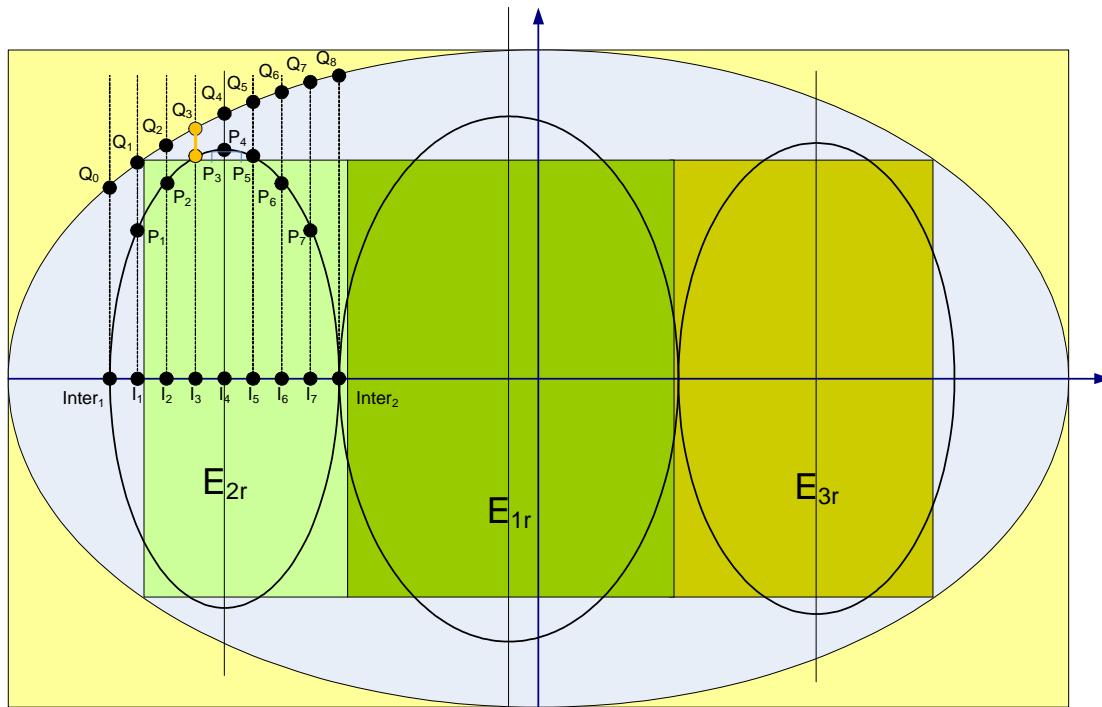


Fig. 133: Algorithme DGC: 11^{ème} étape

Nous adopterons la définition générique suivante pour toutes les ellipses E_{ir} :

$$[(x - x_{ir})^2 / a_{ir}^2] + [(y - y_{ir})^2 / b_{ir}^2] = 1$$

Comme nous avons construit ces ellipses aux étapes précédentes, nous sommes capables d'instancier l'expression générique pour toutes ces ellipses.

Tout d'abord, par symétrie, nous ne considérons que des points situés au-dessus de l'axe des x (c.-à-d. tels que $y > 0$). Ensuite, pour réduire le temps de calcul, nous choisissons uniquement n points I_u répartis de manière homogène entre les intersections de E_{ir} et de l'axe des x . Les points d'intersection $Inter_1$ et $Inter_2$ ont pour coordonnées génériques :

$$Inter_1(-a_{ir} + x_{ir}, 0)$$

$$Inter_2(a_{ir} + x_{ir}, 0)$$

Ils sont donc précisément identifiés.

³³ Une dizaine de points est un compromis fournissant des résultats corrects.

Nous divisons la distance en x entre $Inter_1$ et $Inter_2$ en $(n-1)$ longueurs égales pour définir les n points I_u . Ils sont donc situés aux coordonnées suivantes :

$$I_u(-a_{ir} + x_{ir} + u [2 a_{ir} / (n - 1)], 0) \quad \text{avec } u = 0 \dots (n-1)$$

ce qui peut se réécrire : $I_u(x_u, 0)$ avec $x_u = -a_{ir} + x_{ir} + u [2 a_{ir} / (n - 1)]$ et $u = 0 \dots (n-1)$

Notons que $I_0 = Inter_1$ et que $I_{n-1} = Inter_2$.

A présent, nous définissons les points P_u de coordonnées $(P_{u,x}, P_{u,y})$, situés sur le périmètre de l'ellipse E_{ir} : $P_u(x_u, E_{ir}(x_u))$ avec $u = 0 \dots (n-1)$

Définissons aussi les points Q_u , de coordonnées $(Q_{u,x}, Q_{u,y})$, comme les points de l'ellipse extérieure E ayant les mêmes abscisses que les points I_u , c.-à-d. $Q_{u,x} = x_u$ et $Q_{u,y} = E(x_u)$

Nous pouvons alors vérifier si les points P_u sont situés à l'intérieur de l'ellipse E en évaluant l'expression : $IE_u = (Q_{u,y} - P_{u,y})$

- Si $IE_u > 0$, alors P_u se trouve à l'intérieur de E
- Si $IE_u = 0$, alors P_u se situe sur le périmètre de l'ellipse E
- Si $IE_u < 0$, alors P_u est à l'extérieur de E .

Définissons à présent IE_{min} comme le minimum des valeurs IE_u , soit $IE_{min} = \min\{IE_u\}$.

Si $IE_{min} > 0$, alors tous les points P_u sont situés à l'intérieur de E .

Cette vérification doit être appliquée pour toutes les ellipses E_{ir} . Nous obtenons donc au final un ensemble de valeurs IE_{min-ir} (= valeur de IE_{min} pour l'ellipse E_{ir}).

Définissons $IE_{min, total}$ comme le minimum des valeurs IE_{min-ir} , soit $IE_{min, total} = \min\{IE_{min-ir}\}$.

Deux cas peuvent se présenter.

1^{er} Cas : $IE_{min, total} < 0$

Au moins un point P_u d'une des ellipses E_{ir} est situé à l'extérieur de E . Le point P'_u le plus critique est celui qui correspond à la valeur $IE_{min, total}$ car c'est celui qui est le plus éloigné³⁴ de son point Q'_u associé.

Par conséquent, nous connaissons à la fois l'ellipse E_{ir} concernée et les points P'_u et Q'_u critiques.

Il nous faut donc à nouveau réduire la taille des ellipses E_{ir} . Nous appliquerons donc à toutes les ellipses $E_{i,r}$ le facteur de réduction : $R_x = R_y$

Pour des raisons identiques à celle citée précédemment, les facteurs R_x et R_y seront des valeurs constantes.

Après avoir appliqué le facteur de réduction selon les deux axes x et y à toutes les ellipses, nous répétons l'opération désormais bien connue de translation pour rendre les ellipses tangentes selon l'axe des x . Ensuite, nous recommençons la 11^{ème} étape.

³⁴ Selon la direction de l'axe y

2^{ème} Cas : $IE_{\min, \text{total}} \geq 0$

Cette configuration est visualisée sur la Fig. 133. Tous les points P_u de toutes les ellipses E_{ir} soit se trouvent à l'intérieur de E , soit sont tangents³⁵ à E .

La solution est satisfaisante mais on ne peut exclure qu'elle puisse être améliorée en termes d'occupation de l'espace disponible dans E . Comme cet objectif constitue le but premier de l'algorithme DGC, il nous faut évaluer si nous pouvons encore agrandir les ellipses tout en les gardant à l'intérieur de E .

Nous matérialisons ce critère en imposant que l'une des ellipses soit 'presque' tangente à l'ellipse extérieure E .

Le caractère de 'presque tangence' sera défini par l'expression : $0 < IE_{\min, \text{total}} < \epsilon$

La borne ϵ étant fixée à une valeur donnée³⁶.

Si nous définissons le point P_u' comme le point critique qui est le plus proche³⁷ de son point Q_u associé, nous observons qu'il s'agit aussi de celui qui correspond à la valeur $IE_{\min, \text{total}}$.

De façon symétrique au 1^{er} cas, nous appliquerons à toutes les ellipses E_{ir} un facteur de dilatation : $D'_x = D'_y$ qui sera égal à une valeur constante. En pratique, une valeur comprise entre 1,01 et 1,1 est satisfaisante³⁸.

Notre démarche générale reste valable et après la dilatation des ellipses E_{ir} selon les deux axes x et y , nous translatons les ellipses résultantes selon l'axe des x pour les rendre tangentes.

La 11^{ème} étape sera réitérée jusqu'à ce que la valeur $IE_{\min, \text{total}}$ soit inférieure à l'espacement ϵ choisi.

Nous avons réalisé une série d'expérimentations avec l'algorithme DGC pour déterminer s'il permettait d'améliorer l'utilisation de l'espace d'affichage disponible. Comme on peut aisément le remarquer en comparant les Fig. 134 et Fig. 135, c'est effectivement le cas. Ces figures illustrent les différences entre les résultats obtenus par les deux algorithmes pour un jeu de données simulées incluant 500 nœuds et 8 niveaux de profondeur.

Cependant, l'algorithme DCG présente encore des éléments que l'on pourrait améliorer.

- Si un nœud ne possède qu'un seul enfant, les ellipses de ces deux nœuds se confondent car l'algorithme est 'trop performant' et il fait disparaître l'espace entre les ellipses. Notons que cet inconvénient peut être annulé en activant une fonction d'espacement comparable à celle introduite dans l'algorithme initial (cf. Fig. 108).
- L'algorithme DCG génère l'apparition d'un effet de bord que nous avons appelé la problématique des nœuds cousins. Cette question est discutée en détail dans une section ultérieure (cf. p. 217).
- L'algorithme DGC ne solutionne pas la problématique de la distortion inter-niveaux (cf. p. 200).
- Par nature, l'algorithme DGC nécessite plus de calculs que l'algorithme simple qui ne contient pas d'étapes itératives. Il exige donc plus de temps pour afficher la vue ellimap à l'écran. Cette différence de temps peut être imperceptible ou clairement pénalisante selon la structure et la taille de la structure hiérarchique à afficher.
- Enfin, il faut noter que l'algorithme DGC n'est pas applicable aux ellimaps circularisées.

³⁵ Notons que le cas de tangence est très rare. Nous ne le mentionnons qu'à des fins de rigueur mathématique.

³⁶ A titre indicatif, nous avons fixé dans nos travaux la valeur de ϵ à 0,1.

³⁷ Selon la direction de l'axe des y

³⁸ A titre d'information, notre prototype a été finalement configuré avec la valeur $D'_x = D'_y = 1,01$.

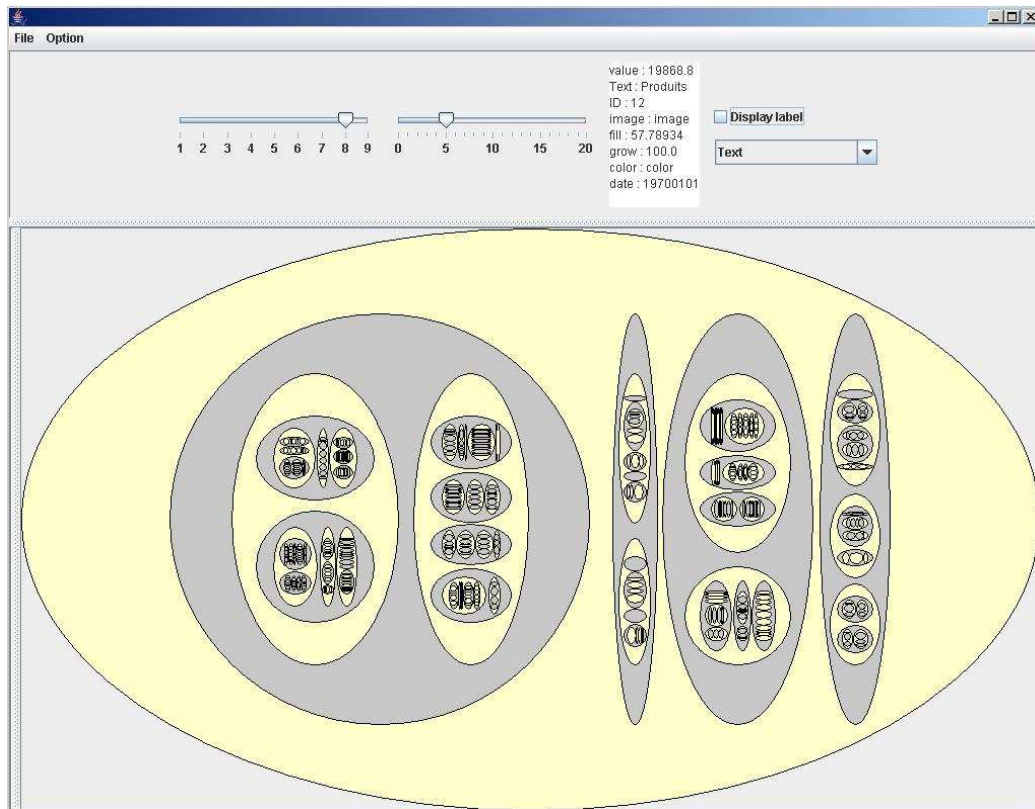


Fig. 134: Algorithme avec division dépendante

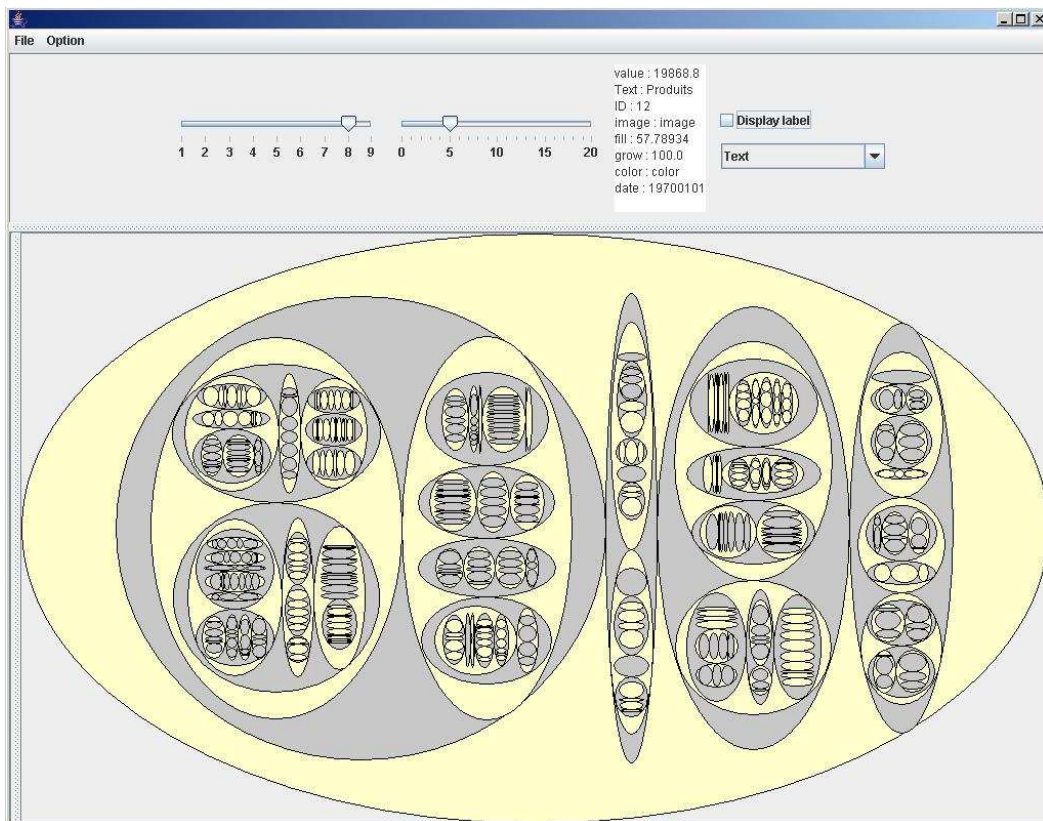


Fig. 135: Algorithme DGC

5.2.1. Problématique des nœuds cousins

Dans l'algorithme DCG, la détermination des facteurs de déformation géométrique génère un effet de bord pour les nœuds appartenant au même niveau hiérarchique mais dépendant de branches différentes de la hiérarchie (problématique des nœuds cousins, cf. Fig. 136).

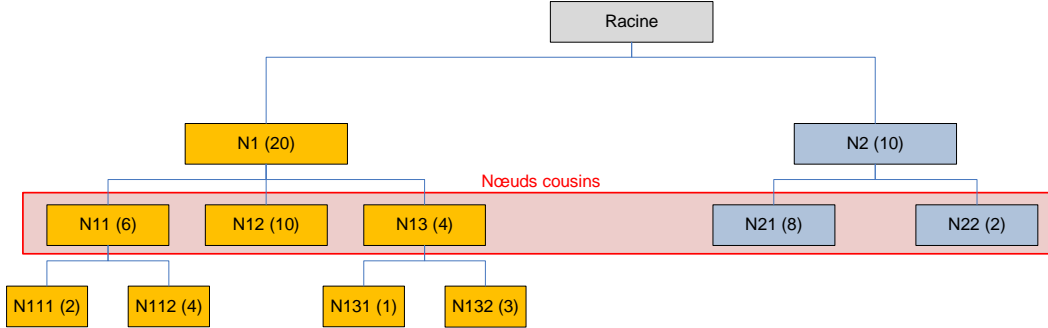


Fig. 136: Problématique des nœuds cousins

En effet, dans la version initiale de l'algorithme de placement des ellipses (cf. p. 189), le rapport des surfaces des ellipses d'un niveau à l'autre est constant quel que soit le niveau hiérarchique considéré (cf. p. 198). Ceci a pour conséquence que les aires de nœuds cousins sont comparables entre elles.

Par contre, dans l'algorithme optimisé DCG, le rapport des surfaces entre les ellipses inscrites et l'ellipse extérieure dépend de la structure de la hiérarchie et de la pondération des nœuds. Cette propriété constitue d'ailleurs l'objectif fondamental de l'optimisation. Néanmoins, cette caractéristique a pour conséquence que si elle est appliquée telle que présentée précédemment à tous les niveaux hiérarchiques, les aires des nœuds cousins ne seront plus comparables car elles auront été déformées avec des facteurs différents pour chaque branche hiérarchique. Des nœuds ayant la même pondération et situés au même niveau hiérarchique pourront donc être représentés par des ellipses de surface différente.

D'un point de vue général, la solution à ce problème consiste à imposer le même ratio entre la surface SE_n de toute ellipse de niveau n et la somme des surfaces des ellipses qui lui sont inscrites au niveau $(n+1)$. Dans l'exemple illustré par la Fig. 136, cette règle se matérialise en imposant que :

$$R_1 = (SE_{N11} + SE_{N12} + SE_{N13}) / SE_{N1} = (SE_{N21} + SE_{N22}) / SE_{N2}$$

et que

$$R_2 = (SE_{N111} + SE_{N112}) / SE_{N11} = (SE_{N131} + SE_{N132}) / SE_{N13}$$

Il faut remarquer que rien n'impose que R_1 soit égal à R_2 car nous ne nous préoccupons ici que de la comparaison de nœuds cousins, c.-à-d. situés au même niveau hiérarchique. Nous savons par ailleurs que la comparaison inter-niveaux pose d'autres difficultés (cf. p. 200).

Pour résoudre le problème des nœuds cousins, nous devons leur imposer le même rapport R_i pour chacune des déformations. Cette contrainte se matérialise dans l'algorithme DCG en modifiant les 9^{ème}, 10^{ème} et 11^{ème} étapes. Le principe est assez simple, il suffit de modifier le calcul des valeurs Y_{ir} et $IE_{min, total}$. Au lieu de calculer ces valeurs en prenant en compte l'ensemble des ellipses associées aux nœuds frères du nœud courant, il faut les calculer en considérant l'ensemble des ellipses associées aux nœuds cousins.

Il faut néanmoins remarquer que cette modification implique que les ellipses associées aux différentes branches de la hiérarchie ne sont alors plus calculées indépendamment les unes des autres, ce qui augmente le temps de calcul nécessaire pour aboutir à la solution. Une alternative intéressante pourrait consister à n'annuler le problème des nœuds cousins que pour les premiers niveaux de profondeur affichés à l'écran. Les niveaux plus profonds conserveraient la distorsion. Cependant comme la surface réelle qu'ils occupent est plus

petite, il est moins probable que l'utilisateur s'y réfère pour comparer exactement les nœuds ayant des pondérations proches.

5.3 Analyse de l'avantage de l'algorithme optimisé.

Le but avoué de l'algorithme optimisé DCG est de diminuer l'espace soi-disant 'perdu' dans l'algorithme initial de l'ellimap. Bien que nous expliquions dans une section ultérieure qu'il est inexact de considérer cet espace inter-ellipse comme absolument inutile (cf. p. 228), nous adopterons néanmoins ce point de vue dans un premier temps car il présente l'avantage de simplifier la discussion. Cette section constitue une généralisation de la discussion présentée au point 5.1 (p. 198).

Avant tout, il faut garder à l'esprit que nous limitons notre discussion au cas des visualisations de hiérarchies basées sur des inclusions successives de formes géométriques.

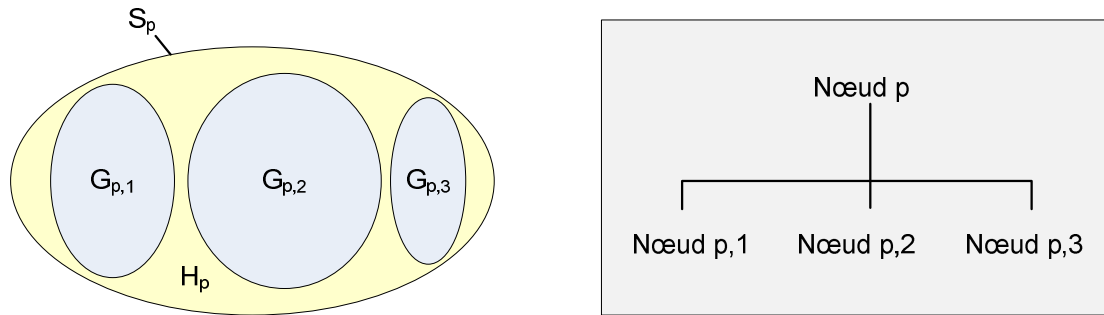


Fig. 137: Illustration du calcul de l'occupation de l'espace d'affichage

Nommons S_p la surface associée à la représentation graphique du nœud p (cf. Fig. 137).

S_p se décompose en trois éléments : la surface $G_{p,i}$ des formes géométriques³⁹ associées aux n nœuds enfants du nœud p , la surface H_p ⁴⁰ destinée à représenter la structure hiérarchique et la surface $L_{p,i}$ dédiée à afficher, dans une zone non incluse dans les surfaces $G_{p,i}$ ou H_p , des informations sur le nœud⁴¹.

$$S_p = H_p + \sum_{i=1}^n (G_{p,i} + L_{p,i})$$

En pratique, les informations sur les nœuds sont souvent affichées dans les surfaces $G_{p,i}$ ou H_p ou via des affichages dynamiques qui, par définition, n'occupent pas d'espace en permanence. Nous pouvons donc négliger le terme $L_{p,i}$ pour obtenir l'expression :

$$S_p = H_p + \sum_{i=1}^n G_{p,i} = H_p + G_p$$

La partie de l'espace d'affichage disponible qui n'est pas utilisée pour représenter les nœuds de la hiérarchie est habituellement considérée comme de l'espace perdu. Dans cette optique, nous pouvons définir le taux de remplissage $T_{R,p}$ associé à un nœud p au moyen de l'expression :

$$T_{R,p} = G_p / S_p = (S_p - H_p) / S_p$$

³⁹ Rectangles dans un *treemap* ou ellipses dans une *ellimap*.

⁴⁰ Il s'agit dans la plupart des cas de l'espace laissé libre entre les formes géométriques associées aux nœuds.

⁴¹ Par exemple des étiquettes (*labels*) affichant le nom des nœuds.

Pour une technique qui exploite parfaitement le principe de remplissage d'espace, le taux de remplissage $T_{R,p} = 1$ pour tout nœud p .

Par contre, pour les techniques qui dévient de l'espace entre les représentations des nœuds pour visualiser la structure de la hiérarchie, il faut étudier en détail la nature du terme H_p .

Selon l'algorithme retenu pour positionner les éléments (*layout algorithm*), ce terme peut prendre des formes très diverses.

- Par exemple, nous avons démontré précédemment (cf. p. 198) que pour l'ellimap avec l'algorithme initial, la surface totale des ellipses de niveau $(q+1)$ était directement proportionnelle à la surface de l'ellipse de niveau q . Notons que si nous appliquons un facteur f de réduction des surfaces dédiées à chaque nœud (cf. Fig. 108), celui-ci doit également être pris en compte dans le calcul du terme H_p .
- Dans le cas d'un *treemap* avec bordure d'épaisseur constante, le terme H_p inclura une composante dépendant du périmètre des rectangles représentant les nœuds.
- Si nous considérons le cas de l'ellimap avec algorithme optimisé DCG, l'espace H_p sera déterminé de manière complexe par les différentes déformations géométriques appliquées aux ellipses. Dans ce cas, les taux de remplissages $T_{R,p}$ varient à chaque niveau car ils dépendent de la structure de la hiérarchie et du poids relatif des nœuds.

Par ailleurs, afin de pouvoir comparer des techniques de visualisation selon le critère du taux de remplissage, nous ne devons pas comparer un seul nœud p mais l'ensemble des nœuds de la hiérarchie.

Obtenir une expression mathématique du taux de remplissage global T_R relatif à l'ensemble des niveaux pour un algorithme tel que DCG n'est pas tâche aisée. D'un point de vue théorique, il nous apparaît difficile d'obtenir dans ce cas une formulation générique comparable à celle de la section 5.1.1 (p. 198). Nous avons donc adopté une approche expérimentale.

Nous avons intégré dans le module de visualisation de notre prototype une fonctionnalité qui calcule, pour chaque nœud p le taux de remplissage $T_{R,p}$. Nous avons donc pu disposer via ce procédé empirique d'une base de comparaison des différentes approches.

A titre d'illustration, nous avons analysé le cas de notre jeu de test composé de 500 nœuds (cf. Fig. 134 et Fig. 135). Comme le taux de remplissage $T_{R,p}$ ne s'applique pas aux feuilles de la hiérarchie, il reste, dans notre exemple, 113 nœuds parents à considérer.

Nous avons calculé la moyenne μ et l'écart-type σ des valeurs $T_{R,p}$ obtenues (cf. Table 15).

Table 15: Comparaison des taux de remplissage pour le jeu de données avec 500 nœuds

	Algorithme initial	Algorithme DCG
Moyenne $T_{R,p}$	0,5	0,6304
Ecart-type $T_{R,p}$	0	0,0747

Nous sommes évidemment conscients du caractère particulier de ces résultats. Les valeurs obtenues ne peuvent pas être directement généralisées car elles dépendent de l'instance du jeu de données considérée. Néanmoins, elles indiquent une tendance générale en phase avec l'objectif de conception de l'algorithme DCG : mieux utiliser l'espace disponible que l'algorithme initial des ellimaps. Une première indication montre qu'un gain moyen de l'ordre de 13 % est obtenu par l'algorithme DGC. Cette amélioration peut sembler limitée mais n'oublions pas qu'il s'agit d'une loi exponentielle qui s'applique à chaque changement de niveau.

Nous avons voulu mieux comprendre ce que représente ce gain de 13%. Dans cette optique, nous avons considéré que le gain observé de 13% est une première approximation utilisable de la loi réelle.

Sous ces conditions, la Table 16 témoigne du gain obtenu en termes d'espace d'affichage disponible pour une hiérarchie à 10 niveaux. Au 5^{ème} niveau de profondeur, l'espace disponible est trois fois plus important pour l'algorithme DCG que pour l'algorithme initial. Au 10^{ème} niveau, ce ratio s'élève à un facteur 10 !

Table 16: Comparaison du gain d'espace d'affichage entre l'algorithme initial et DGC

Niveau hiérarchique	Algorithme initial	Algorithme DGC	Ratio entre les surfaces d'affichage disponibles
0	100	100	1
1	50	63,04	1,2608
2	25	39,740416	1,58961664
3	12,5	25,05235825	2,00418866
4	6,25	15,79300664	2,526881062
5	3,125	9,955911385	3,185891643
6	1,5625	6,276206537	4,016772184
7	0,78125	3,956520601	5,064346369
8	0,390625	2,494190587	6,385127902
9	0,1953125	1,572337746	8,050369259
10	0,09765625	0,991201715	10,14990556

Si on considère que l'espace d'affichage total disponible est un écran d'ordinateur avec une résolution 1280 x 1024 pixels, la Table 17 renseigne le nombre de pixels disponibles pour afficher l'ensemble des nœuds de niveau n . Nous avons considéré que 80% de l'écran étaient disponibles pour afficher le rectangle extérieur initial de l'ellimap. Notons également que l'ellipse associée à la racine est identique dans les deux cas et que son aire occupe 78,54 % de l'espace disponible (cf. p. 198).

Table 17: Gain d'espace d'affichage en pixels

Niveau hiérarchique	Algorithme initial : pixels disponibles	Algorithme DGC : pixels disponibles
80 % de l'écran	1.048.576	1.048.576
Racine (0)	823.551	823.551
1	411.775	519.166
2	205.887	327.282
3	102.943	206.319
4	51.471	130.063
5	25.735	81.992
6	12.867	51.687
7	6.433	32.583
8	3.216	20.540
9	1.608	12.949
10	804	8.163

En toute rigueur, il faudrait estimer le gain de l'algorithme DCG sur base d'un large ensemble de hiérarchies construites aléatoirement. Par manque de ressources, nous n'avons pu entreprendre une telle opération. Néanmoins, les éléments dont nous disposons nous indiquent que l'algorithme DCG occupe mieux l'espace d'affichage disponible que l'approche initiale.

5.4 Représentation du nœud racine de la hiérarchie

Afin d'augmenter encore l'exploitation de l'espace d'affichage, nous avons réfléchi à la pertinence de visualiser explicitement le nœud racine de la hiérarchie. En effet, la connaissance de la surface de l'ellipse associée n'est pas utilisée en pratique. Par contre, sur base des règles de calcul approximatif élaborées dans la section précédente, nous pouvons estimer le gain généré par cette fonctionnalité. L'avantage est indéniable car nous réduisons la 'perte' d'espace au niveau le plus élevé et donc le plus important en termes quantitatifs.

Nous proposons donc une variante des ellimaps dans laquelle la zone rectangulaire disponible au total pour l'affichage représente la racine de la hiérarchie (cf. Fig. 138). Les premières ellipses visualisées correspondent au niveau 1.

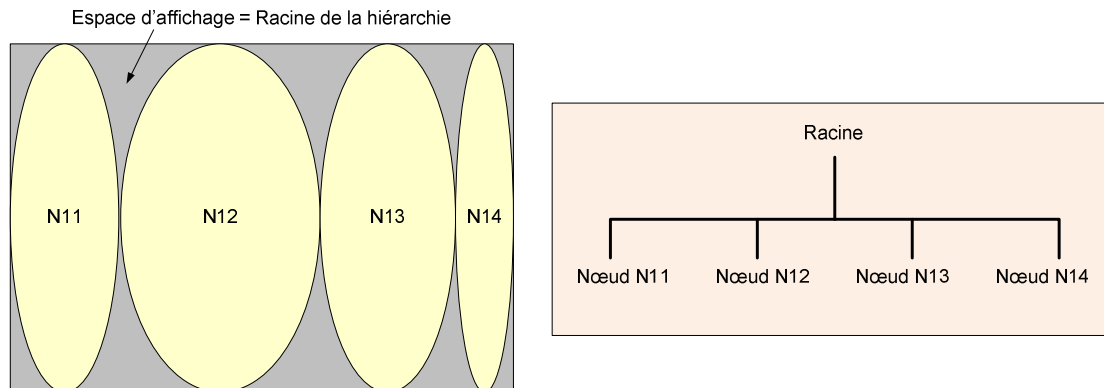


Fig. 138: Suppression de l'affichage du nœud racine comme une ellipse

La Table 18 indique le gain d'espace consécutif à cette variante des ellimaps. Notons que cette adaptation est applicable à tous les algorithmes de positionnement des ellipses (p.e.x division dépendante, circularisés, DGC).

Au premier niveau, les ellipses occupent 78,54% de la surface du rectangle associé à la racine (cf. Fig. 117).

Table 18: Gain d'espace d'affichage en pixels avec la suppression de l'ellipse racine

Niveau hiérarchique	Algorithme initial : pixels disponibles	Algorithme DGC : pixels disponibles
80 % de l'écran Rectangle associé à la racine	1.048.576	1.048.576
1	823.551	823.551
2	411.775	519.166
3	205.887	327.282
4	102.943	206.319
5	51.471	130.063
6	25.735	81.992
7	12.867	51.687
8	6.433	32.583
9	3.216	20.540
10	1.608	12.949

6. Ellimaps-réservoirs

6.1 Problématique

Nous abordons ici une problématique peu traitée dans la littérature relative à la visualisation de hiérarchies : la représentation graphique des structures de données dont les nœuds sont associés à des ressources consommables.

Pourtant, cette propriété spécifique se rencontre dans de multiples situations pratiques :

- la visualisation du temps alloué à une tâche et du temps déjà utilisé pour réaliser cette tâche ;
- la visualisation du nombre de messages total dans un répertoire et du nombre de messages lus ;
- la visualisation de l'espace disponible sur une plate-forme électronique pour chacun des espaces de travail et celle de l'espace effectivement occupé par chacun d'entre eux ;
- la visualisation d'un poste budgétaire et celle des sommes déjà engagées sur ce poste.

Deux valeurs numériques sont donc associées aux nœuds : la première correspond à une capacité ou à une quantité de ressources disponibles et la seconde représente la quantité de ressources consommées.

Ressources disponibles	Ressources consommées
Temps alloué à une tâche	Temps mis pour réaliser la tâche
Nombre de messages à lire	Nombre de messages lus
Espace disponible	Espace occupé
Poste budgétaire	Somme dépensée
Chiffres d'affaire planifié	Chiffre d'affaires effectif

Table 19: Exemples de ressources consommables

Plus précisément, nous pouvons reformuler ce problème comme suit.

Nous sommes confrontés à la problématique de visualiser des paires de valeurs numériques dont l'une représente une fraction de l'autre, ce qui peut s'exprimer comme suit :

$\{u_i, v_i\}$ tq $u_i = k \cdot v_i$ avec $k \in [0, 1]$

Une paire de valeurs est associée à chacun des nœuds de la hiérarchie H .

Au point de vue de la visualisation, cette problématique se décompose en deux parties : visualiser les paires de valeurs $\{u_i, v_i\}$ et les affecter à une représentation de données hiérarchiques. Nous traiterons les deux sous-problèmes séquentiellement.

6.2 Le concept de jauge

Si on ne considère pas leur affectation aux nœuds d'une structure hiérarchique, la représentation de paires de valeurs telles que $\{u_i, v_i\}$ est un problème très courant en visualisation d'information.

Les multiples variantes de jauges sont les solutions les plus courantes pour répondre à cette question. Dans cette optique, la théorie de la sémiologie graphique de Bertin (1998) présente un intérêt particulier. Elle explique que la position dans le plan et la taille sont probablement les meilleures variables visuelles pour supporter la comparaison de données quantitatives. Pour illustrer cette règle, on peut citer les exemples de l'histogramme qui utilise une des dimensions du plan pour représenter la valeur des éléments d'un ensemble donné ou du *treemap* dans lequel la taille des rectangles visualise le poids des nœuds d'une hiérarchie.

Il apparaît donc naturel qu'un grand nombre de jauges se basent également sur ces variables. D'une manière générale, elles consistent alors à visualiser les valeurs u_i et v_i comme des quantités géométriques telles que des longueurs (cf. jauge A), des angles (cf. jauge B) ou des surfaces (cf. jauge C). La valeur u_i correspondant à la fraction k de v_i est représentée comme la fraction équivalente de la quantité géométrique choisie (cf. Fig. 139).

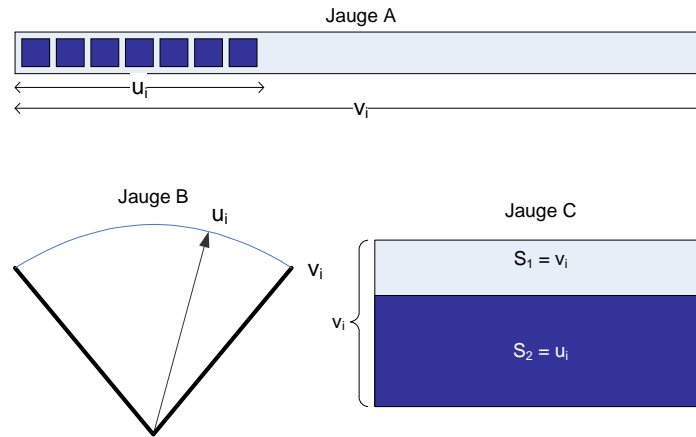


Fig. 139: Exemples de jauges

Néanmoins, les jauges peuvent aussi être basées sur d'autres variables visuelles, telles que la couleur.

Ceci nous amène à aborder un cas particulier de la problématique générale du pseudo-coloriage (*pseudocoloring*) qui étudie la manière de visualiser une gamme de valeurs en utilisant une séquence de couleurs (cf. Ware, 2004, p. 127). Dans le problème qui nous préoccupe, nous nous focalisons sur les séquences ordinales de couleurs car les différentes valeurs possibles u_i et la valeur v_i sont numériques et ordonnables. Dans cette configuration, Ware (cf. Ware, 2004, p. 129) conseille d'utiliser une séquence de couleurs qui croît de manière monotone selon un ou plusieurs canaux de couleurs opposées. Selon la théorie '*Opponent Process Theory*', il existe trois axes perceptuels selon lesquels les couleurs élémentaires sont associées en paires : 'noir – blanc', 'rouge – vert' et 'jaune – bleu' (cf. Ware, 2004, p. 110). D'un point de vue cognitif, cette théorie porte sur des processus neuronaux qui se produisent dans le cerveau. Elle ne doit donc pas être confondue avec la théorie de Young-Helmoltz qui est acceptée à l'heure actuelle et qui postule l'existence dans l'œil humain de trois sortes de récepteurs (appelés cônes) qui répondent aux couleurs : rouge, vert et bleu (cf. Gregory, 2000, p. 164). Sur base de la théorie '*Opponent Process Theory*', une manière immédiate de représenter une séquence de valeurs (cf. Ware, 2004, p. 129) consiste donc à leur associer une couleur dans la gamme des gris (canal 'noir – blanc') ou dans la gamme de couleurs 'rouge – vert' ou 'jaune – bleu'. Ware précise qu'il est également possible de créer un ordonnancement monotone en utilisant la propriété de saturation de la couleur ou en utilisant des couleurs positionnées le long d'une droite dans l'espace des couleurs opposées. Par ailleurs, il précise aussi que s'il est important de montrer des détails, le canal de luminance (c.-à-d. le canal 'noir – blanc') est à préférer car il possède une grande capacité pour traduire de l'information à haute fréquence de variation spatiale.

A titre d'exemple, on peut colorier les rectangles imbriqués d'un *treemap* sur une échelle bichromatique 'vert – rouge' (cf. Fig. 140). Dans une telle approche, il est important de préciser que la couleur est utilisée comme seul moyen de visualiser la portion k de ressources consommées. Nous aborderons plus tard l'utilisation de la couleur en vue d'un double codage de la variable k .

En étudiant plus en détail l'exemple de la Fig. 140, nous pouvons observer deux limites de cette approche. En premier lieu, nous remarquons qu'il n'est pas aisé d'identifier exactement à quel pourcentage de consommation correspond une couleur donnée. Par exemple, il n'est

pas aisé d'identifier la couleur exacte correspondant à une consommation de 50% de la ressource disponible.

Ensuite, nous remarquons que deux variables visuelles différentes sont utilisées pour coder deux concepts intimement reliés. La quantité absolue de ressources disponibles v_i est représentée par la taille des rectangles et le niveau de consommation u_i par leur couleur. Pour effectuer des tâches de comparaison des valeurs de consommation absolue, l'utilisateur doit donc combiner deux échelles conceptuellement différentes : la surface et la couleur. Par exemple, dans la Fig. 140, comparer deux nœuds frères selon la valeur absolue de ressources consommées demande, pour chaque nœud, de traduire mentalement la couleur en pourcentage puis de multiplier ce pourcentage par la surface du rectangle associé. Dans une telle approche, il n'est pas aisé de comparer les valeurs de consommation relatives à un petit rectangle rouge et un grand rectangle vert.

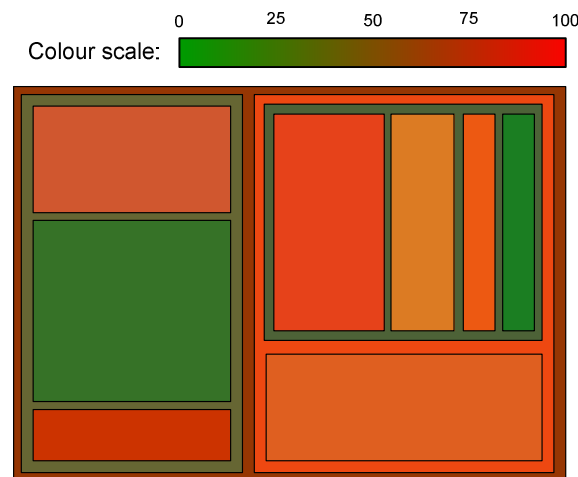


Fig. 140: Treemap avec jauge basée sur une échelle bichromatique

La couleur peut par contre s'avérer être une variable intéressante quand la valeur de u_i est représentative de concepts qualitatifs tels qu'un niveau de danger, de risque ou de retard. Dans ce cas, la variable associée à une quantité géométrique est représentative de la proportion de ressources consommée et la couleur code une seconde information : la zone qualitative dans laquelle cette valeur se situe.

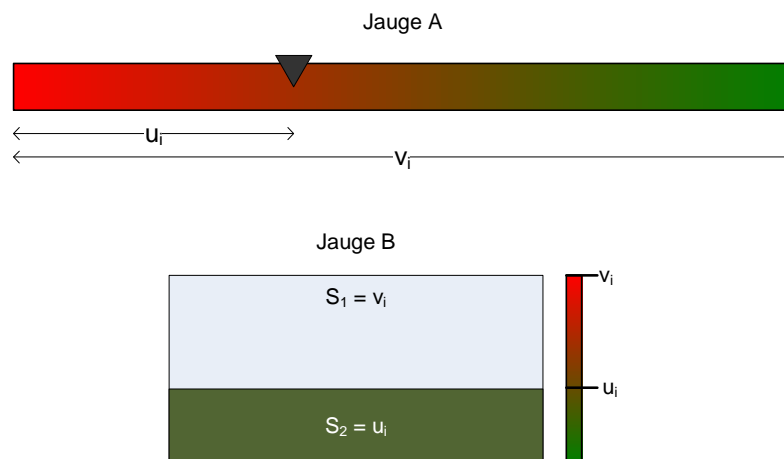


Fig. 141: Exemples de jauges combinant pseudo-coloriage et dimension du plan

La Fig. 141 illustre cette approche. Des jauges basées sur la dimension linéaire (jauge A) ou la surface (jauge B) sont combinées avec du pseudo-coloriage sur une échelle 'rouge - vert'. La couleur sert principalement à attirer l'attention sur des valeurs critiques alors que la dimension linéaire ou la surface sont utilisées pour évaluer la proportion de ressources consommées.

6.3 La transformation de formes géométriques en jauges

Nous avons concentré nos travaux sur les techniques basées sur la loi de clôture car elles présentent l'avantage d'associer les nœuds de la hiérarchie à des formes géométriques dont la surface est représentative d'une valeur numérique. Nous étudions dans cette section la capacité de ces différentes formes géométriques à être transformées en jauges.

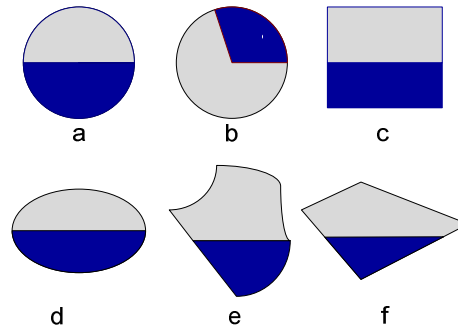


Fig. 142: Transformation de formes géométriques en jauges

La Fig. 142 montre différentes approches pour générer des jauges à partir de formes géométriques. Il apparaît assez nettement que la meilleure forme est probablement le rectangle (c) car l'estimation du niveau de ressources consommées nécessite l'estimation d'une distance linéaire. Cette opération s'avère plus difficile dans le cas du cercle (a) et de l'ellipse (d) car la surface et la hauteur de la zone colorée sont reliées par une fonction non-linéaire. L'estimation du ratio entre la surface colorée et la surface complète est définitivement trop compliquée pour des formes plus complexes telles que celles associées aux Voronoi treemaps [8] (e) ou à l'algorithme RELT (Hao et al., 2007) (f). Enfin, l'utilisation des angles (b) peut aussi représenter une approche digne d'intérêt.

La facilité d'estimation du niveau de ressources consommées n'est pas le seul facteur qui doit être pris en considération. Comme notre but est d'inclure des jauges dans des techniques de visualisation basées sur l'inclusion successive de formes géométriques, il nous faut aussi considérer cet élément. Dans ce contexte, la représentation par diagrammes en camembert (b) présente des inconvénients car le principe des formes imbriquées entraîne que certaines zones colorées peuvent être presque complètement cachées par les formes qui leur sont superposées.

Enfin, nous souhaitons imposer l'emploi d'une métaphore qui peut être facilement comprise par une grande diversité d'utilisateurs. Pour cette raison, nous avons choisi de limiter notre étude aux jauges qui reposent sur la métaphore de réservoirs de stockage (a, c, d, e, f). Cette métaphore est en effet universelle et ne devrait pas présenter trop de biais dépendant de facteurs individuels ou culturels. Ceci constitue un second argument pour exclure de notre étude les jauges basées sur des secteurs de cercle (b) qui ne sont pas compatibles avec la loi de gravité universelle.

En conclusion, nous sélectionnons les techniques de visualisation basées sur des rectangles, des cercles et des ellipses (a, c, d) pour approfondir notre analyse.

6.4 L'ajout de jauges à des techniques existantes

La technique de visualisation que nous sommes en train de concevoir devra représenter chaque nœud de la hiérarchie comme une jauge. Il ne suffit donc pas d'analyser la capacité d'une forme géométrique à être transformée en jauge mais il est aussi indispensable de s'intéresser à la qualité de la figure globale obtenue. Par conséquent, nous étudions dans cette section dans quelle mesure les figures produites par des *treemaps*, des *treemaps* circulaires (Wetzel, 2004), des *bubble trees* (Boardman, 2000) et des ellimaps supportent une telle adaptation. Cette évolution pose deux questions cruciales.

- Les nouveaux éléments visuels représentant les jauges interfèrent-ils avec les formes géométriques imbriquées les unes dans les autres ? (Q1)
- Les jauges sont-elles perçues comme telles dans la figure globale ? (Q2)

6.4.1. Ajout de jauges à un treemap

La Fig. 143 illustre à quoi ressemblerait un *treemap* dans lequel les rectangles seraient transformés en jauges. Dans ce contexte, l'algorithme utilisé pour positionner et dimensionner les rectangles n'est pas très important car il ne modifie pas le fait qu'un *treemap* soit fondamentalement un ensemble de rectangles imbriqués.

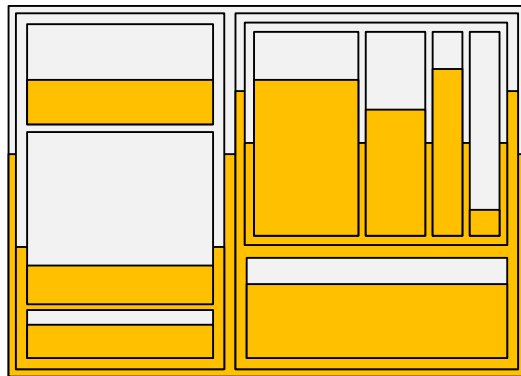


Fig. 143: Ajout de jauges à un treemap

Nous observons que les jauges rectangulaires sont très semblables aux rectangles imbriqués, ce qui peut induire une certaine confusion. En fait, dans cette approche, la technique de visualisation transforme subtilement la nature des données. La représentation de deux valeurs numériques pour un nœud donné est remplacée par la visualisation de deux enfants pour ce nœud. Ceci résulte de l'usage dual d'une même primitive graphique : les rectangles sont utilisés pour visualiser à la fois la structure des données et les jauges.

Nous voyons aussi que la seconde question (Q2) ne reçoit une réponse satisfaisante que pour les feuilles de la hiérarchie. Pour les niveaux supérieurs, le but principal (et une qualité majeure) des *treemaps* qui consiste à optimiser l'occupation de l'espace d'affichage, devient une limitation pour visualiser les jauges.

En fait, comme les jauges sont incluses dans la représentation pour visualiser une seconde variable numérique pour chaque nœud, il n'est pas très surprenant qu'une technique optimisée pour représenter des nœuds avec une seule valeur numérique ne soit pas la meilleure. Pour ajouter les jauges, nous avons besoin d'espace libre et cet espace 'perdu' est exactement ce que les *treemaps* cherchent à minimiser.

6.4.2. Ajout de jauges à un treemap circulaire

Les *treemaps* circulaires peuvent aussi être transformés pour inclure des jauges (cf. Fig. 144). Par rapport aux *treemaps*, cette technique répond mieux à la première question (Q1). En effet, les lignes horizontales sont clairement perçues comme différentes des cercles. Le principe de continuité de la '*Gestalt*' théorie peut contribuer à expliquer ce phénomène. En effet, il indique que '*nous sommes plus susceptibles de construire des entités visuelles à partir d'éléments visuels qui sont lisses et continus qu'à partir d'autres qui contiennent des changements abrupts de direction*' (Ware, 2004). Les niveaux de consommation ne sont donc pas confondus avec les cercles qui représentent la structure des données.

Concernant la seconde question (Q2), la Fig. 144 montre bien que cette approche s'accommode parfaitement de la métaphore des réservoirs de stockage. Les cercles partiellement colorés rappellent la section transversale d'un réservoir cylindrique et peuvent sans effort être

identifiés comme des indicateurs de consommation partielle. De plus, l'espace libre entre les cercles est plus grand que dans le cas des *treemaps* et les jauges sont donc plus visibles, quel que soit le niveau hiérarchique considéré.

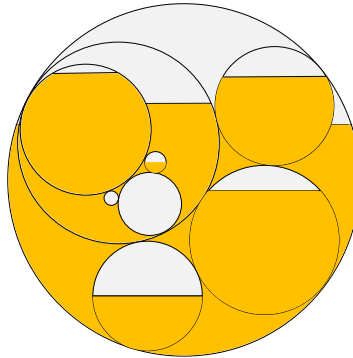


Fig. 144: Ajout de jauges à un treemap circulaire

Cette approche présente cependant deux défauts. Le premier a déjà été discuté et concerne la relation non-linéaire entre la surface et la hauteur de la zone colorée. Le second découle de la disposition des cercles inscrits les uns dans les autres. La disposition non structurée de la représentation peut perturber l'utilisateur. Aucun axe de référence n'est disponible pour comparer les nœuds, même pour ceux appartenant au même niveau hiérarchique. De ce point de vue, la visualisation plus structurée proposée par le *treemap* est une meilleure solution. De plus, dans des circonstances favorables, le niveau de consommation relatif peut être facilement comparé pour des nœuds frères car les rectangles sont alignés selon l'axe horizontal (cf. partie supérieure droite de la Fig. 143).

6.4.3. Ajout de jauges à un bubble tree

Les *bubble trees* peuvent inclure des jauges (cf. Fig. 145) selon un principe similaire aux *treemaps* circulaires. Ils présentent les mêmes avantages concernant les réponses aux questions Q1 et Q2 mais leur disposition plus structurée peut aider à réduire la confusion pour l'utilisateur.

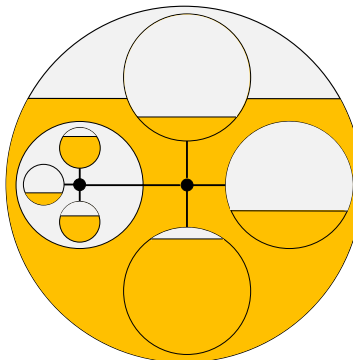


Fig. 145: Ajout de jauges à un bubble tree

Malheureusement, ils souffrent d'une lacune majeure: la taille des cercles n'est pas proportionnelle au poids des nœuds. Dès lors, ils ne peuvent être employés que pour comparer des niveaux relatifs de consommation des ressources.

6.4.4. Ajout de jauges à une ellimap

Les ellimaps peuvent être considérées comme un compromis entre les *treemaps* et les *treemaps* circulaires. Elles apparaissent donc comme de bons candidats potentiels pour supporter l'inclusion de jauges. La Fig. 146 illustre cette approche.

Les ellimaps combinent différents points positifs des approches précédentes. D’une part, elles sont composées de formes courbes imbriquées comme les *treemaps* circulaires. Cet élément offre des avantages en termes de perception visuelle des niveaux de consommation. D’autre part, elles héritent des *treemaps* de certains algorithmes de positionnement des ellipses qui les rendent plus structurées que les *treemaps* circulaires. Par rapport aux *treemaps*, l’occupation moins efficace de l’espace d’affichage est transformée en avantage car cela offre de l’espace pour afficher les jauges.

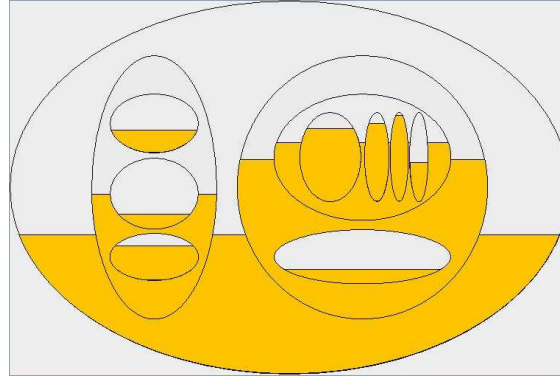


Fig. 146: Ajout de jauges à une ellimap

En conclusion, sur base de ces différents arguments, nous avons choisi d’ajouter des jauges à la technique des ellimaps.

6.5 Le principe des ellimaps-réservoirs

Pour commencer, il peut être utile de rappeler que dans les ellimaps que nous avons décrites jusqu’à présent, la surface des ellipses est représentative d’une valeur numérique associée aux nœuds de la hiérarchie. L’approche la plus naturelle consiste donc à associer le couple de valeurs $\{u_i, v_i\}$ à des surfaces. Plus précisément, la surface totale S_E de l’ellipse E sera associée à la valeur totale v_i et la valeur partielle S_k sera visualisée par une partie de la surface S_E égale à $k.S_E$ (cf. Fig. 147).

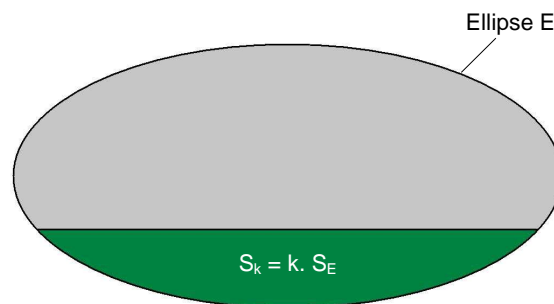


Fig. 147: Visualisation d’une jauge dans une ellipse

Nous avons discuté précédemment du caractère assez intuitif de cette représentation graphique. Elle fait allusion à la métaphore d’une citerne d’une capacité v_i remplie jusqu’au niveau u_i . On peut facilement imaginer que l’ellipse constitue une coupe transversale de la citerne. Nous nommerons donc cette technique : ‘ellimap-réservoir’ ou ‘*storage-ellimap*’ en anglais.

D’un point de vue conceptuel, les ellimaps-réservoirs (cf. Fig. 146) présentent plusieurs propriétés intéressantes.

Tout d’abord, la distinction entre les ellipses formées de lignes courbes et les droites qui indiquent le niveau de consommation d’une ressource est clairement perceptible. L’utilisation de primitives graphiques différentes (courbe vs. droite) permet d’éviter la confusion entre

l'objet graphique qui représente le nœud (c.-à-d. l'ellipse) et l'objet graphique qui délimite le niveau (c.-à-d. la droite).

Deuxièmement, elles reposent sur une métaphore universelle, celle d'un récipient rempli de liquide. La compréhension de l'information perçue ne devrait donc pas poser de problèmes à la majorité des utilisateurs.

Enfin, elles offrent une réponse à la limitation des ellimaps classiques en termes d'occupation de l'espace d'affichage. En effet, dans le cas des ellimaps-réservoirs, l'espace libre entre les ellipses imbriquées est utilisé pour visualiser de l'information, à savoir le taux de consommation de ressources associées aux nœuds.

6.6 Développement géométriques

La concrétisation pratique de cette approche nécessite une réflexion au niveau géométrique.

Avant tout, il est nécessaire d'isoler les cas particuliers $k = 0$, $k = 0,5$ et $k = 1$, pour lesquels la solution est triviale.

Ensuite, nous devons distinguer deux configurations : $k \in]0, \frac{1}{2}[$ et $k \in]\frac{1}{2}, 1[$

Nous commencerons par traiter le cas où la fraction k représente moins de 50%. Dans ce contexte, nous prendrons la Fig. 149 comme référence pour notre réflexion.

La géométrie nous fournit la formule de l'aire S_k du segment d'ellipse :

$$S_k = 0,5 a \cdot b \cdot (r - \sin r) \quad \text{avec l'angle } r \text{ exprimé en radians}$$

Nous imposons que $S_k = k \cdot S_E$

Or, l'aire de l'ellipse $S_E = \pi \cdot a \cdot b$

Par conséquent, nous pouvons écrire : $S_k = k \cdot \pi \cdot a \cdot b$

Nous pouvons évaluer les deux expressions de S_k : $k \cdot \pi \cdot a \cdot b = 0,5 a \cdot b \cdot (r - \sin r)$

Après simplification, nous obtenons l'équation suivante :

$$F_1(r) \equiv \sin r - r + 2 \cdot k \cdot \pi = 0$$

En d'autres termes, pour une valeur donnée de k , la solution r^* de cette équation fournit la valeur de l'angle r qui définit la surface S_k égale à la fraction k de l'aire totale S_E . La fonction $F_1(r)$ est strictement décroissante et possède une seule solution r^* sur l'intervalle $r \in]0, \pi[$

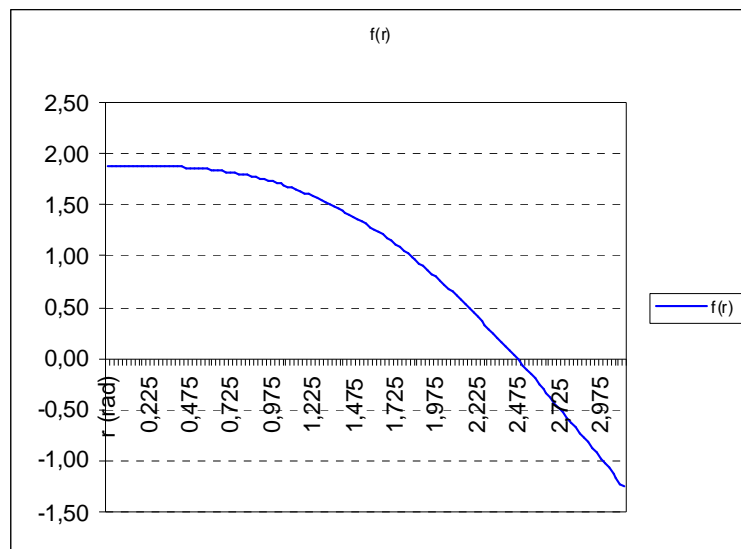


Fig. 148: Graphe de la fonction $F_1(r)$ avec $k = 0,3$

A titre d'exemple, la Fig. 148 illustre le graphe de $F_1(r)$ pour $k = 0,3$.

Malheureusement la valeur de r^* ne peut être exprimée sous forme analytique et il nous faudra donc la calculer numériquement. Cette question sera traitée plus tard.

La Fig. 149 constituera notre point de référence pour nos développements géométriques.

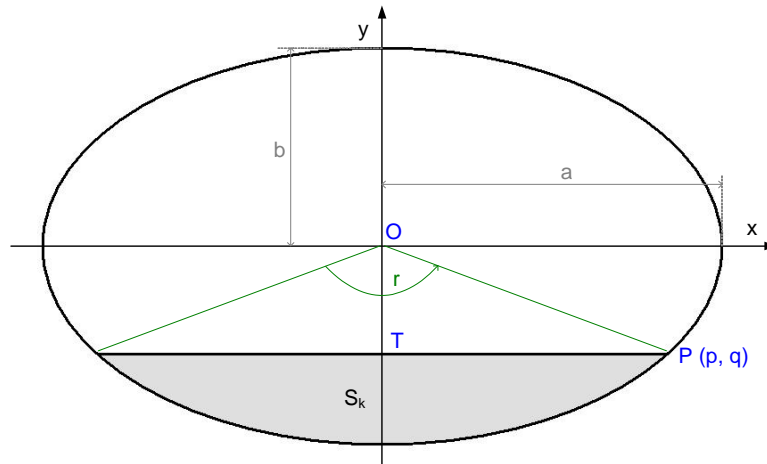


Fig. 149: Calcul de la surface colorée S_k ($k < 0,5$)

Dès lors que la valeur de r^* est connue, nous pouvons calculer les coordonnées (p, q) du point P qui facilitent le dessin de la figure.

Dans le triangle OPT, nous pouvons écrire $p/q = \tan(r^*/2)$, pour autant que nous considérons les valeurs absolues de p et q .

Vu que le point P appartient à l'ellipse E, il satisfait à l'équation :

$$y = -(b/a) \sqrt{(a^2 - x^2)} \text{ et donc } q = -(b/a) \sqrt{(a^2 - p^2)}$$

Si nous remplaçons q par sa valeur dans cette expression, nous obtenons :

$$p / \tan(r^*/2) = -(b/a) \sqrt{(a^2 - p^2)}$$

Elevons les deux membres de l'égalité au carré : $a^2 \cdot p^2 = b^2 \cdot \tan^2(r^*/2) \cdot (a^2 - p^2)$

Après quelques développements, nous obtenons la valeur de p et q :

$$p = a \cdot b \cdot \tan(r^*/2) \cdot (a^2 + b^2 \cdot \tan^2(r^*/2))^{-1/2} \quad \text{et} \quad q = -a \cdot b \cdot (a^2 + b^2 \cdot \tan^2(r^*/2))^{-1/2}$$

A présent, nous devons examiner le cas où $k \in] 1/2, 1 [$

Par complémentarité, nous savons que : $S_E = S_k + S_c$

Nous pouvons donc écrire : $S_k = S_E - 0,5 a \cdot b \cdot (r - \sin r)$

Comme $S_k = k \cdot S_E$ cette équation peut s'exprimer comme suit : $k \cdot S_E = S_E - 0,5 a \cdot b \cdot (r - \sin r)$

ou encore $k \cdot \pi \cdot a \cdot b - \pi \cdot a \cdot b + 0,5 a \cdot b \cdot (r - \sin r) = 0$

soit finalement

$$F_2(r) \equiv \sin r - r + 2 \pi \cdot (1 - k) = 0$$

Comme pour le cas précédent, pour une valeur de k donnée, la solution r^{**} de cette équation fournit la valeur de l'angle r qui définit la surface S_k égale à la fraction k de l'aire totale S_E . Cette équation possède aussi une seule solution sur l'intervalle $r \in] 0, \pi [$ Ici encore, la

valeur de r^{**} ne peut malheureusement pas être exprimée sous forme analytique et il faudra la calculer numériquement.

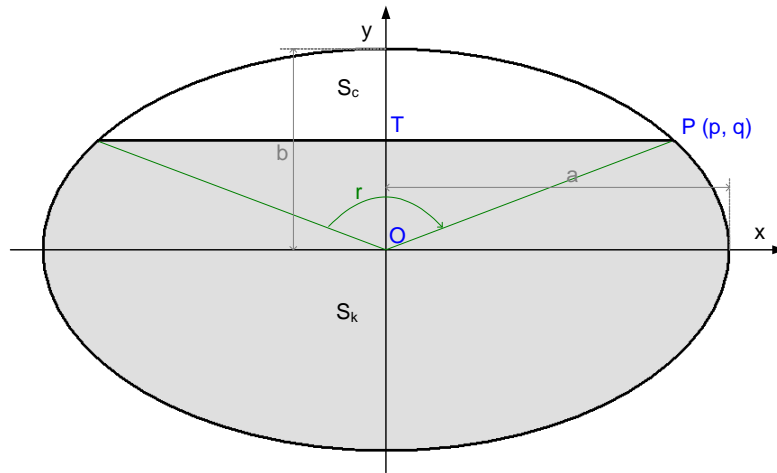


Fig. 150: Calcul de la surface colorée S_k ($k > 0,5$)

Une fois que la valeur de r^{**} est connue, nous pouvons calculer les coordonnées du point P qui facilitent le dessin de la figure.

Le raisonnement géométrique explicité pour le cas précédent est toujours valable. Dès lors, les valeurs de p et q sont données par les expressions ci-dessous :

$$p = a \cdot b \cdot \operatorname{tg}(r^{**}/2) \cdot (a^2 + b^2 \cdot \operatorname{tg}^2(r^{**}/2))^{-1/2} \quad \text{et} \quad q = a \cdot b \cdot (a^2 + b^2 \cdot \operatorname{tg}^2(r^{**}/2))^{-1/2}$$

A ce stade de notre développement, la question de la résolution numérique des équations reste pendante. Le temps est venu de la traiter.

Tout d'abord, il peut être utile de faire remarquer que les deux équations correspondent en fait à la même équation générique :

$$F(r) \equiv \sin r - r + 2 \cdot t \cdot \pi = 0$$

Le paramètre t prend ses valeurs dans l'intervalle $] 0, \frac{1}{2} [$

En effet, si $k \in] 0, \frac{1}{2} [$ alors $t \in] 0, \frac{1}{2} [$ et si $k \in] \frac{1}{2}, 1 [$ alors $t = (1-k)$ et $t \in] 0, \frac{1}{2} [$

Différentes méthodes numériques sont disponibles pour trouver la racine r^* de l'équation qui nous préoccupe, notamment la méthode par dichotomie, la méthode *Regula Falsi*, la méthode de la sécante ou la méthode de Newton. Il ne s'agit pas d'un problème particulièrement complexe.

Le tableau ci-dessous fournit la valeur calculée de r^* pour quelques valeurs du paramètre t.

t	r^* ($\epsilon < 10^{-4}$)
0,10	1,6268
0,20	2,1131
0,30	2,4908
0,40	2,8248

Table 20: Liste de couples de valeurs (t, r^*)

Dans le but d'optimiser le temps de traitement pour construire une ellimap-réservoir, il pourrait être intéressant d'éviter le calcul numérique de r^* . Dans ce contexte, la connaissance d'une fonction qui approxime la relation entre r^* et t s'avérerait particulièrement utile.

Nous sommes donc confrontés au problème d'interpolation numérique suivant : trouver une fonction qui interpole le mieux possible un ensemble de n points P_i de coordonnées connues.

L'interpolation polynômiale (aussi appelée interpolation de Lagrange) constitue une première réponse à cette question. Elle consiste à trouver un polynôme de degré $(n-1)$ qui passe par n points P_i connus. Malheureusement, elle génère une mauvaise approximation aux bornes de l'intervalle quand le degré du polynôme d'interpolation augmente (cf. phénomène de Runge). Cet inconvénient peut être néanmoins réduit si on choisit les racines des polynômes de Tchebychev comme points P_i au lieu de points équidistants.

L'interpolation par splines cubiques est une seconde approche applicable à notre problématique. Elle présente l'intérêt de ne pas utiliser des polynômes de degré élevé et de ne pas être sujette au phénomène de Runge. De plus, elle minimise la courbure de la fonction d'interpolation. L'idée de base consiste à trouver $(n-1)$ polynômes de degré 3 qui sont définis sur les $(n-1)$ sous-intervalles $[P_j, P_{j+1}]$ et qui se raccordent sans à-coups. La fonction globale d'interpolation est donc une fonction définie par morceaux. Plus précisément, nous adopterons la méthode des splines cubiques naturels qui annule les dérivées secondes des polynômes aux extrémités $\{P_0, P_{n-1}\}$ de l'intervalle d'interpolation.

La première étape de l'interpolation consiste à préciser l'intervalle sur lequel la fonction sera interpolée. Dans notre cas, nous choisissons l'intervalle $[0,01, 0,49]$. Il couvre les variations de l'aire S_k comprises entre 1% et 99%, ce qui nous semble être une précision suffisante dans le cas des ellimaps-réservoirs. Il est en effet très probable que des variations inférieures au pourcent ne soient pas perceptibles sur la représentation graphique de l'ellimap.

Le tableau ci-dessous fournit les coordonnées des 11 points P_i que nous considérerons dans le processus d'interpolation.

	t	r^* ($\epsilon < 10^{-4}$)
P_0	0,01	0,7290
P_1	0,05	1,2690
P_2	0,10	1,6268
P_3	0,15	1,8915
P_4	0,20	2,1131
P_5	0,25	2,3099
P_6	0,30	2,4908
P_7	0,35	2,6612
P_8	0,40	2,8248
P_9	0,45	2,9842
P_{10}	0,49	3,1102

Table 21: Liste des points d'interpolation P_i

La méthode des splines cubiques naturels nous fournit alors les 10 polynômes de degré 3 qui définissent la fonction d'interpolation. Rappelons que le polynôme $S_j(t)$ est défini sur l'intervalle $[P_t, P_{t+1}]$.

	Coefficient en t^3	Coefficient en t^2	Coefficient en t	Constante
$S_0(t)$	-885.1632174370965	26.554896523112895	14.650712182668222	0.5807225517384436
$S_1(t)$	719.8006401685169	-214.18968211772915	26.687941114710327	0.38010206953774184
$S_2(t)$	-95.01975714452317	30.256437076182937	2.243329195319116	1.194922466850782
$S_3(t)$	60.27838840957791	-39.627728423162544	12.72595402022094	0.6707912256056912
$S_4(t)$	0.3062035062063759	-3.644417481139629	5.529291831816354	1.1505687048326634
$S_5(t)$	9.696797565605127	-10.687363025688693	7.290028217953622	1.0038406726545577
$S_6(t)$	4.1066062313650535	-5.656190824872626	5.7806765577088015	1.1547758386790399
$S_7(t)$	3.4767775089388953	-4.99487066632516	5.549214502217188	1.1817797451530612
$S_8(t)$	2.7862837328763543	-4.166278135050112	5.217777489707168	1.2259713468210638
$S_9(t)$	3.37329246389194	-4.958739921921152	5.5743852937991365	1.1724801762072687

Table 22: Définition des splines cubiques d'interpolation

La Fig. 151 illustre le graphe de la fonction d'interpolation composée des 10 polynômes cubiques. Nous pouvons donc maintenant calculer facilement la valeur de r^* pour une valeur donnée de t .

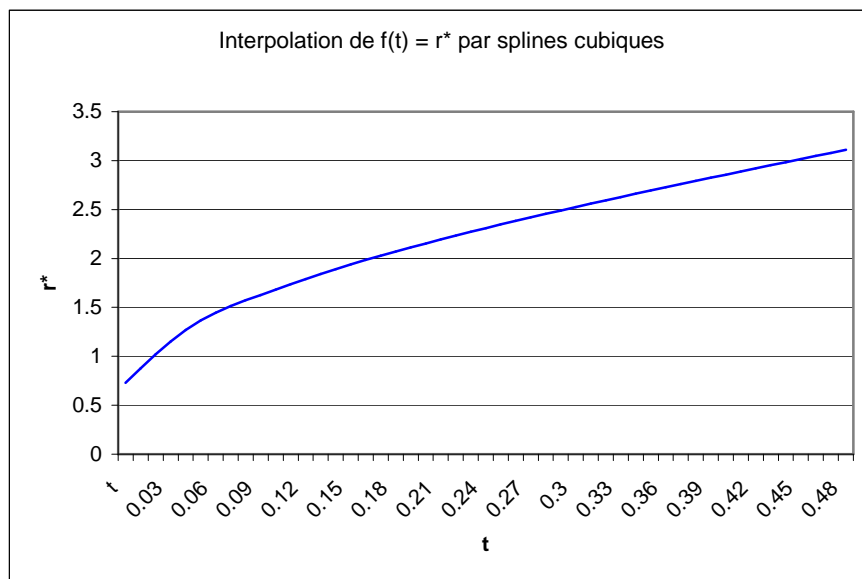


Fig. 151: Graphe de la fonction d'interpolation $f(t) = r^*$ par splines cubiques

Par exemple, si $t = 0,21$ l'intervalle concerné est $[P_4, P_5]$ et le polynôme à utiliser est $S_4(t)$. Nous obtenons donc :

$$r^* = 0.3062035062063759 t^3 - 3.644417481139629 t^2 + 5.529291831816354 t + 1.1505687048326634$$

soit $r^* = 2,153836929$ rad

Nous pouvons vérifier que $S_k / S_E = (r^* - \sin r^*) / (2 \cdot \pi) = 0,2099$ soit $\sim 0,21$.

Cet exemple correspond en réalité à deux alternatives : soit $k = 0,21$ ou $k = 0,79$.

Dans le premier cas, nous obtenons comme coordonnées du point P :

$$p = 1,85742543 \cdot a \cdot b \cdot (a^2 + 3,45002923 \cdot b^2)^{-1/2}$$

et

$$q = -a \cdot b \cdot (a^2 + 3,45002923 \cdot b^2)^{-1/2}$$

Dans le second cas, nous obtenons :

$$p = 1,85742543 \cdot a \cdot b \cdot (a^2 + 3,45002923 \cdot b^2)^{-1/2}$$

et

$$q = a \cdot b \cdot (a^2 + 3,45002923 \cdot b^2)^{-1/2}$$

Prenons l'exemple d'une ellipse E centrée en (0,0) dont les dimensions sont telles que $a = 5$ et $b = 3$.

L'aire S_k correspondant à une proportion de 21 % de l'aire totale sera délimitée par une droite déterminée par les points $(-3,72146, -2,00356)$ et $(3,72146, -2,00356)$.

A présent, l'algorithme qui permet de générer le dessin d'une ellipse partiellement colorée est complètement connu. Pour construire une ellimap-réservoir, il suffit de l'appliquer aux algorithmes de construction des ellimaps décrit précédemment (cf. Fig. 152⁴²).

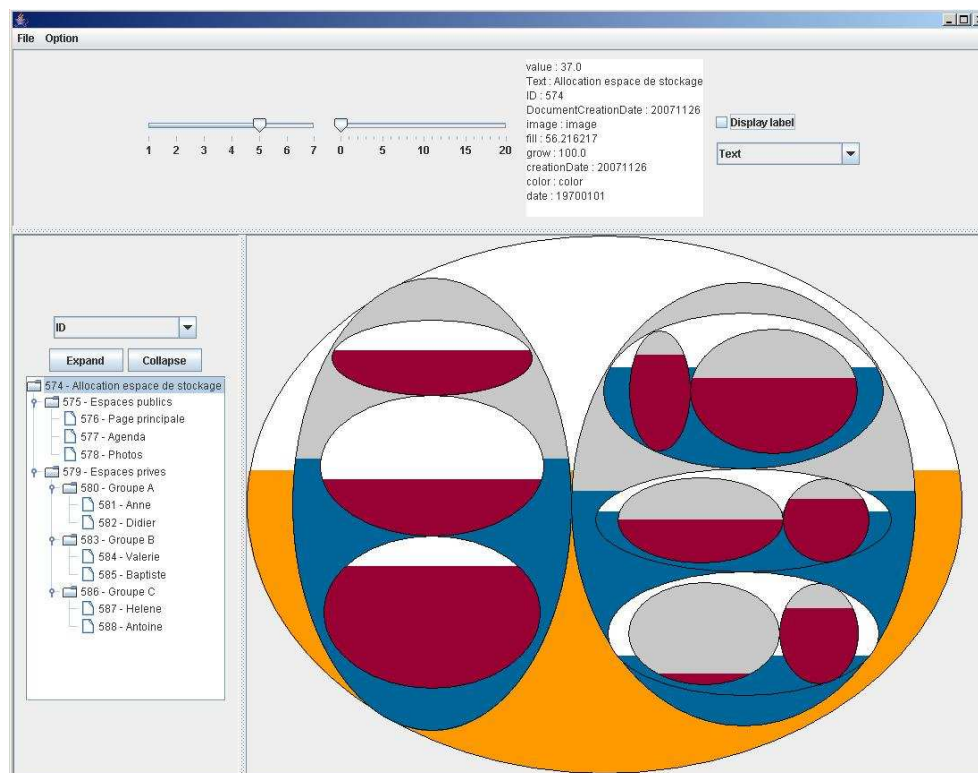


Fig. 152: Ellimap-réservoir (avec algorithme DGC)

⁴² Notons que dans l'exemple de la Fig. 152, les surfaces représentant les niveaux de consommation adoptent une couleur dépendant de leur position dans la hiérarchie (feuilles → bordeaux, nœuds intermédiaires → bleu, racine → orange). La couleur de fond des ellipses adopte quant à elle la stratégie classique de l'alternance des couleurs à chaque niveau de profondeur (racine → blanc, niveau 1 → gris, niveau 2 → blanc...). Ce choix n'est cependant qu'une des multiples options possibles.

D'un point de vue plus général, la technique des ellimaps-réservoirs propose une nouvelle approche pour améliorer les techniques de visualisation basées sur la loi de clôture. Pour ces méthodes, un certain consensus existe dans la littérature pour définir l'idée d'optimisation comme un processus visant à atteindre la meilleure occupation possible de l'espace d'affichage. Cette optique conduit le concepteur à réaliser un compromis entre deux éléments : les formes géométriques imbriquées et l'espace libre résiduel qui les sépare. La perception de la structure du jeu de données est améliorée si l'espace libre augmente mais c'est au détriment du nombre de nœuds qui peuvent être affichés.

Jusqu'à présent, l'objectif d'optimisation a souvent été interprété comme celui d'afficher le plus grand nombre possible de nœuds dans un espace d'affichage fixe. Par conséquent, les chercheurs ont essayé des moyens de minimiser l'espace libre entre les formes géométriques associées aux nœuds. Dans cette optique, l'espace libre est d'ailleurs qualifié d'espace perdu. Ce point de vue est indubitablement justifié quand il s'agit de visualiser de grands ensembles de données. Nous l'avons également adopté pour élaborer l'algorithme DGC.

Cependant, une approche complémentaire est possible. Elle consiste à réfléchir aux moyens d'exploiter plutôt que de réduire cet espace libre. Les ellimaps-réservoirs démontrent qu'en plus de mettre en évidence la structure de la hiérarchie, l'espace libre peut aussi être utilisé pour visualiser une seconde variable pour chaque nœud. Dans cette perspective, il n'est plus pertinent de l'appeler 'espace perdu'. On peut rappeler dans ce contexte que cet espace a déjà été utilisé pour afficher des étiquettes, pour supporter un codage par couleur ou pour permettre certaines interactions avec la visualisation (cf. p.ex. logiciel Treemap HCIL 4.1.1., 2004).

Nous pensons que cette nouvelle manière de considérer le concept d'optimisation dans les méthodes basées sur la loi de clôture peut ouvrir de nouveaux champs de recherche.

Nous attirons aussi l'attention sur le fait que les *treemaps* ne représentent que deux informations sur les nœuds intermédiaires de la hiérarchie visualisée : la position dans la structure et le poids des nœuds. L'approche des ellimaps-réservoirs ouvre des perspectives pour visualiser des informations supplémentaires sur ces nœuds intermédiaires (éventuellement autres que le niveau de consommation de ressources).

Nous rappelons également que l'usage de surfaces pour représenter les nœuds exige de comparer les surfaces selon deux dimensions qui sont en relation linéaire (rectangles) ou non-linéaire (cercles, ellipses). Pour cette raison, nous pensons que cette approche est plutôt destinée à supporter des comparaisons approximatives plutôt que l'identification de valeurs exactes. Néanmoins, nous sommes convaincus que la figure globale peut utilement aider l'utilisateur dans une phase d'exploration des données.

Chapitre 8 : Ellimaps : Evaluation

« Observer attentivement, c'est se rappeler distinctement »

E. A. Poe

1. La problématique de l'évaluation

1.1.1. Cadre général

Nous avons décrit dans les chapitres précédents différentes techniques pour visualiser des informations associées à des plates-formes électroniques de coopération (cf. *ambient flags*, *clock view*, *ellimaps*). Afin de connaître leur potentiel applicatif réel, il était nécessaire de procéder à leur évaluation. Malheureusement, comme nous ne disposons pas des ressources pour évaluer en détail chacune d'entre elles, nous avons dû opérer un choix. La technique des ellimaps nous semble la plus prometteuse d'un point de vue général. C'est d'ailleurs celle que nous avons le plus développée tant d'un point de vue théorique que de celui de son implémentation logicielle. Nous avons donc décidé de mettre en œuvre la campagne d'évaluation la plus approfondie pour cette technique de visualisation. Néanmoins, les autres techniques ont également fait l'objet d'évaluations mais d'une ampleur plus limitée.

La question de l'évaluation de logiciels s'avère être d'une complexité et d'une difficulté particulièrement élevée même si on se limite, comme dans ce chapitre, à l'évaluation de la composante relative aux interfaces utilisateurs et à l'interaction homme-ordinateur.

Selon Sears (2003), trois approches sont actuellement considérées comme des standards en matière d'évaluation d'interfaces homme-machine : l'approche basée utilisateur (*user-based evaluation*), l'approche basée sur une inspection (*inspection-based evaluation*) et l'approche basée sur des modèles (*model-based evaluation*).

Nous avons choisi une classification plus détaillée basée sur le but de l'évaluation, sur la nature de la personne qui évalue le logiciel, sur la localisation respective de l'évaluateur et du sujet et sur la méthode de collecte des informations.

Il faut noter que la nature de la tâche à évaluer influence évidemment la méthodologie à retenir. Nous y consacrerons une section détaillée ultérieurement.

a. But de l'évaluation

Au niveau méthodologique, on opère habituellement une distinction entre évaluation formative et sommative. Dumas (2003) explique que : '*Historiquement, les professionnels en facteur humain ont fait une distinction entre les mesures formatives et sommatives. Un test formatif est fait tôt dans le développement pour contribuer à la conception d'un produit ; un test sommatif est réalisé tard dans le développement pour évaluer la conception du produit.*'

Dans une autre perspective, Lewis (2006b) s'interroge sur les tests d'utilisabilité et il précise qu'ils peuvent se focaliser sur des mesures au niveau des tâches (*task-level measurements*) ou sur la découverte de problèmes (*problem discovery*). Dans le premier cas, il s'agit d'une évaluation sommative (*summative testing*) et dans le second d'une évaluation formative (*formative testing*).

Dans le même ordre d'idées, Shneiderman et Plaisant (2005, p. 148) proposent que les tests d'utilisabilité peu coûteux (*discount usability tests*) reposant sur un petit nombre de participants (3 à 6) soient utilisés comme évaluation formative et que des méthodes plus chères le soient pour l'évaluation sommative.

Pour résumer, le but d'une évaluation peut être d'améliorer le logiciel testé pendant son développement. Il s'agit alors d'en identifier les principales faiblesses et de valider certaines décisions (tests de diagnostic). Cet objectif est associé aux évaluations formatives. Dans

d'autres cas, l'évaluateur cherche à obtenir des informations sur un logiciel terminé afin d'évaluer par exemple le degré de satisfaction qu'il procure à ses utilisateurs. Il peut également viser à comparer deux logiciels ou deux versions d'une interface homme-machine (test comparatifs). Ces tests seront regroupés sous l'appellation d'évaluation sommative.

La Table 23 résume les différences principales entre évaluation formative et sommative.

Table 23: Distinction entre évaluations formatives et sommatives

Evaluation formative	Evaluation sommative
- tôt dans le développement	- tard dans le développement
- participe à la conception	- évalue la conception
- découverte de problèmes	- mesures au niveau des tâches
- tests rapides et peu coûteux	- tests plus élaborés et plus lourds à mettre en œuvre

b. Nature de la personne qui évalue

Deux grandes approches peuvent être distinguées à ce niveau selon que l'évaluation est réalisée par un expert ou par un utilisateur final.

Les **évaluations par experts** se basent souvent sur une inspection. Cockton et al. (2003) définissent cette approche comme suit : « *Une méthode d'inspection de l'utilisabilité (Usability Inspection Method – UIM) est appliquée à un système interactif pour évaluer son utilisabilité. Un analyste inspecte le système (ou une spécification ou un croquis) et rapporte sur son utilisabilité.* »

Les différentes méthodes d'évaluation basées sur une inspection par un analyste peuvent être regroupées en deux grandes approches. La première consiste à évaluer la conformité du système testé avec un ensemble de règles. Selon le niveau d'abstraction de celles-ci, on les qualifie de principes, d'heuristiques ou de règles de style (*style rules*). Dans la seconde approche (appelée *walk-through*), une analyse pas à pas d'une interaction hypothétique avec le système est réalisée. Enfin, outre les méthodes reposant sur une inspection, les experts peuvent également conduire des évaluations en se basant sur des modèles.

Les méthodes reposant sur des heuristiques (*heuristic evaluation*) impliquent d'évaluer une interface par rapport à des 'règles d'or'. Les violations de ces règles sont identifiées et considérées comme des problèmes d'utilisabilité. Les heuristiques de Nielsen (1994) figurent parmi les plus connues. A titre d'exemple, l'une d'entre elles précise que le statut du système doit être visible.

Les lignes de conduite (*guidelines*) fournissent des conseils à un niveau d'abstraction plus bas que les heuristiques. Elles indiquent par exemple comment aider l'utilisateur à encoder une date dans un champ. Elles constituent des ensembles qui peuvent regrouper plusieurs centaines de règles (cf. p.ex *Guidelines* de Smith et Mossier, 1986). Dès lors, une évaluation basée sur des lignes de conduite peut difficilement les examiner une par une. Elles constituent par contre un cadre qui peut notamment aider l'évaluateur à élaborer des recommandations.

La répétition cognitive (Cockton et al., 2003) (*cognitive walk-through*) est une méthode d'inspection procédurale. L'évaluateur choisit un ensemble de tâches à analyser. Ensuite, les analystes réalisent chaque tâche pas à pas en répondant à différentes questions sur base de preuves empiriques, de leur expérience ou de preuves scientifiques. Une réponse négative à une des questions est le signe d'un problème potentiel d'utilisabilité. Cette méthodologie évalue si l'utilisateur peut élaborer des objectifs et choisir les actions appropriées pour les atteindre à partir de sa connaissance préalable et des signaux présents dans le système (*system cues*). Le succès des méthodes de répétition cognitive repose notamment sur le choix des tâches analysées.

La répétition heuristique (*heuristic walk-through*) proposée par Sears (1997) combine l'utilisation d'heuristiques et la répétition cognitive. En premier lieu, une liste de tâches prioritaires (tâches fréquentes ou critiques) est élaborée. Ensuite, l'évaluateur explore ces tâches en utilisant un ensemble de questions suscitant la réflexion. Ces questions sont dérivées de l'approche '*Cognitive Walk-through*'. Puis, l'évaluateur explore le système complet en utilisant ces mêmes questions et des heuristiques. Selon Sears, la répétition heuristique identifie plus de problèmes réels que la répétition cognitive.

En résumé, les méthodes d'évaluation basées sur des inspections restent, malgré certaines lacunes, un bon complément aux autres types d'approches. Elles conduisent notamment à réduire le nombre de problèmes d'utilisabilité du système avant de conduire un test basé sur des utilisateurs, ce qui permet de mieux exploiter les ressources rares (c.-à-d. les utilisateurs de test). Il nous faut cependant mentionner la revue critique de ces différentes méthodologies réalisée par Gray et Salzman (1998) qui ont montré que plusieurs évaluations basées sur l'une ou l'autre de ces méthodes souffraient de lacunes au niveau méthodologique et que leurs conclusions pouvaient être mises en doute.

Les évaluations basées sur des modèles peuvent également être associées à des experts. Elles regroupent l'ensemble des méthodes reposant sur un modèle de comment un être humain utiliserait le système testé pour obtenir, par calcul ou par simulation, des mesures de l'utilisabilité prévue (Kieras, 2003). Le but de cette approche est donc d'obtenir des informations sur l'utilisabilité d'une application avant d'implémenter un prototype ou de réaliser des tests avec des utilisateurs.

Kieras (2003) explique que les modèles (appelés *engineering* ou *analytic models*) sont basés sur une description détaillée du design proposé et une analyse détaillée des tâches. Les modèles expliquent comment l'utilisateur va accomplir les tâches en interagissant avec l'interface proposée et ils utilisent des théories psychologiques et des données paramétriques pour générer des métriques de l'utilisabilité prévue. Une fois que le modèle est construit, les prédictions sur l'utilisabilité sont obtenues par calcul ou simulation.

On peut donc simuler un raffinement progressif de l'interface utilisateur en modifiant des éléments du modèle et en évaluant leur portée sur l'utilisabilité. De plus, le modèle lui-même résume l'interface et il peut être inspecté en détail pour identifier ses forces et ses faiblesses. Enfin, dans certains cas, des parties de modèle peuvent être réutilisées car elles constituent des modules représentatifs d'interactions élémentaires.

En général, les évaluations basées sur un modèle sont complétées par des évaluations basées sur des utilisateurs. En effet, d'une part les modèles disponibles actuellement ne couvrent pas tous les aspects de l'utilisabilité et d'autre part le processus de modélisation est nécessairement imparfait et il doit être associé à des tests empiriques.

Les évaluations par des utilisateurs finaux représentent la seconde grande famille d'approches pour mettre en œuvre des tests de logiciels. Dans cette approche, les utilisateurs participent de manière directe au processus d'évaluation.

Avant de décrire certaines de ces méthodes, nous attirons l'attention sur deux biais potentiels associés aux évaluations avec des utilisateurs finaux : le fait que les participants changent parfois d'attitude quand ils savent qu'ils sont observés et la tendance des observateurs à vouloir voir ce qu'ils veulent voir.

Selon Dumas (2003), l'évaluation par des utilisateurs finaux est en général mis en œuvre par une (ou une combinaison) des trois approches suivantes :

- des questionnaires administrés aux utilisateurs,
- l'observation d'utilisateurs,
- des tests empiriques d'utilisabilité.

Il peut être utile de présenter sommairement les trois méthodes.

La première méthode repose sur le remplissage de questionnaires par un utilisateur final. Les questionnaires peuvent porter sur des aspects généraux (p.ex. satisfaction de l'utilisateur,

envie d'utiliser le logiciel en pratique) ou plus précis (p. ex. lisibilité d'une zone d'affichage, ordonancement des éléments d'un menu).

Deuxièmement, l'observation des utilisateurs (dans leur environnement naturel) peut également constituer une méthode d'évaluation. Dans ce cas, l'évaluateur observe comment l'utilisateur se sert du logiciel et il enregistre ce qui se passe. Au niveau méthodologique, cette approche présente l'inconvénient que l'observateur ne peut en aucune manière contrôler la survenance des événements. Il est totalement passif, contrairement à une expérience en laboratoire par exemple.

Les tests empiriques d'utilisabilité constituent le troisième type d'approche.

Nous discuterons d'abord les tests de diagnostic portant sur un seul produit (*diagnostic usability tests*). Nous aborderons ensuite la question des tests d'utilisabilité comparatifs (*comparative usability tests*) ou compétitifs (*competitive usability tests*).

Les tests de diagnostic ont pour objectif d'identifier les problèmes liés à des défauts d'utilisabilité. A ce sujet, une question très discutée dans la littérature concerne le nombre d'utilisateurs nécessaires pour recenser la majorité des problèmes et surtout les plus critiques d'entre eux. Les premières recherches à ce sujet ont été réalisées par Virzi (1990, 1992) qui a montré que 80% des problèmes étaient découverts avec cinq participants au test et que 90% des problèmes étaient identifiés avec dix personnes. Les premiers travaux de Nielsen et Landauer (1993) rejoignent également ce point de vue. Cependant, alors que certains auteurs continuent toujours de confirmer cette conclusion, des voix divergentes se sont élevées au cours du temps pour la mettre en cause. Citons par exemple Molich et al. (1999), Spool et Schroeder (2001) ou encore Woolrych et Cockton (2003). Lewis (2006b) insiste par ailleurs sur la nécessité de revenir aux lois fondamentales de la statistique et de déterminer dans quelles conditions un petit nombre d'utilisateurs est suffisant et dans quelles autres il faut augmenter la taille de l'échantillon.

Un autre aspect important des tests d'utilisabilité est mis en avant par Dumas (2003) : « *La plupart des problèmes d'utilisabilité n'émergent pas de l'analyse des données après le test. Les problèmes sont observés pendant les sessions et sont enregistrés sur des formulaires ou dans des fichiers log.* » Les problèmes apparaissent donc en direct aux yeux des évaluateurs. Les tests d'utilisabilité sont donc de bons outils pour explorer des problèmes mais leur potentiel est limité pour mesurer l'utilisabilité.

Les tests comparatifs entre deux produits ont une toute autre vocation que les tests de diagnostic. Ici il ne s'agit pas de mettre en évidence des lacunes dans le produit mais de comparer plusieurs solutions.

Dumas (2003) présente les deux grandes approches pour les tests comparatifs de produits : '*Dans la littérature sur les méthodes de recherche, une configuration [de test comparatif] dans lequel les participants utilisent tous les produits est appelée 'within-subjects' configuration, alors que dans une configuration 'between-subjects' chaque participant utilise un seul produit.*'

La configuration inter-sujets (*between-subjects*) présente l'avantage d'éviter la contamination entre les produits. Cependant, elle pose également des défis à l'évaluateur. Elle nécessite de rassembler des groupes d'utilisateurs équivalents pour utiliser chaque produit. L'équivalence peut notamment inclure des variables telles que le niveau d'éducation, l'expérience avec l'informatique ou la connaissance préalable des logiciels testés. Comme ce critère est difficile à satisfaire, il est important de disposer de groupes de taille suffisante pour éviter les biais dus à la présence de sujets particulièrement bons ou mauvais (biais appelé '*wildcard effect*').

La configuration intra-sujets (*within-subjects*) élimine l'effet de non-équivalence des groupes de testeurs. Par contre, elle génère l'apparition d'autres difficultés. Il faut mentionner notamment les biais qui apparaissent si l'ordre et la séquence des éléments testés (tâches, questions) sont identiques pour tous les sujets (cf. effet de séquence, Howell, 1998, p. 553). L'effet de fatigue dû à une durée trop longue des tests doit également être considéré.

Dumas (2003) précise que, puisque les comparaisons statistiques en configuration intra-sujets ne sont pas influencées par les inégalités entre les groupes, elles sont statistiquement plus puissantes que celles reposant sur une configuration inter-sujets, ce qui signifie que les évaluateurs ont besoin de moins de sujets pour détecter des différences. Dans le même ordre d'idées, Shneiderman et Plaisant (2005, p. 148) expliquent que la configuration intra-sujets semble être la plus puissante, parce que les participants peuvent faire des comparaisons entre les interfaces en compétition. Ils précisent également que moins de participants sont nécessaires mais que la période de test pour chacun d'entre eux est plus longue. Lewis (2006a, p. 1285) explique également que *'en général, un test intra-sujets conduit à des mesures plus précises des différences entre les produits (exigeant un plus petit nombre de participants pour une précision égale [...]) et donne l'opportunité de collecter de la part des participants des comparaisons subjectives directes des produits.'*

Enfin, il faut tenir compte, comme Lewis (2006a, p. 1285) le rappelle à juste titre, que les méthodes d'analyse statistiques à mettre en œuvre diffèrent dans les deux configurations.

En règle générale, pour qu'un test comparatif soit valide, il doit éviter trois grandes sources de biais potentiels.

- La première concerne la sélection des participants. Il est essentiel que le choix des sujets du test ne favorise pas un des produits. Pour valider cette hypothèse, on peut conduire un test de qualification qui évalue le niveau de connaissances préalables susceptibles d'influencer les résultats du test comparatif. Par exemple, l'éventuelle expérience avec un des produits testés peut biaiser le résultat de la comparaison. En effet, Lewis (2006a, p. 1285) explique qu'une des caractéristiques les plus importantes pour définir un groupe lors d'un test d'utilisabilité est l'expérience spécifique à la fois avec les produits testés et le domaine d'intérêt.
- La seconde est relative à la sélection et à la formulation des tâches à accomplir. Les tâches choisies doivent être typiques de l'utilisation classique du produit et être normales pour les utilisateurs. De plus, la description des tâches à accomplir ne doit pas utiliser un vocabulaire qui favorise un des produits.
- La troisième se situe au niveau des interactions entre l'évaluateur et les sujets pendant le test. Elles doivent être limitées au strict minimum.

c. Localisation de l'évaluation

En ce qui concerne la localisation de l'évaluation, nous avons identifié trois cas possibles :

- évaluation conduite dans un **laboratoire**,
- évaluation réalisée sur le **lieu de travail**,
- évaluation à **distance**.

Dans les deux premières configurations, les évaluateurs et le sujet se rencontrent directement avant, pendant et / ou immédiatement après la séance de test. Il existe dès lors un certain nombre de moyens indirects de contrôle ou de validation des informations collectées. Par exemple, l'évaluateur est certain que le sujet est bien la personne qui a réalisé les tests. De plus, l'évaluateur peut discuter informellement avec le sujet après le test pour connaître son opinion informelle sur le logiciel testé.

Dans la troisième configuration, le sujet ne rencontre pas l'évaluateur. Par exemple, le test est réalisé par le biais d'un formulaire disponible sur Internet. Le principal avantage des tests à distance est leur coût plus faible que les évaluations en laboratoire ou sur le lieu de travail. Par contre, ils présentent l'inconvénient d'offrir moins de moyens de contrôle de la validité des données collectées.

d. Méthode de collecte des informations

Notre dernier axe de différenciation des méthodes d'évaluation concerne le moyen utilisé pour collecter les informations issues des tests. Les différentes méthodes que nous avons recensées sont présentées ci-dessous.

L'enregistrement vidéo des tests est une première option possible. Dans ce cas, le comportement du sujet pendant les tests est filmé et enregistré. Ceci nécessite une infrastructure qui n'est souvent disponible que dans un laboratoire. Cette approche présente l'avantage de pouvoir analyser a posteriori toutes les facettes du comportement du sujet, y compris des éléments qui sont le plus souvent ignorés des autres méthodes, comme par exemple les gestes ou les expressions faciales du sujet. Dans certains cas, il peut aussi être demandé au sujet de commenter a posteriori l'enregistrement du test qui le concerne.

Certaines évaluations se basent sur **l'observation des sujets** en train d'utiliser le logiciel dans un contexte d'utilisation normal. Cette méthode ressemble à l'enregistrement vidéo, à ceci près que c'est l'évaluateur qui fait office de caméra. Tout comme celle-ci, il s'efforce de ne pas interagir avec le sujet mais contrairement à celle-ci, il ne peut tout noter. Par contre, cette méthode ne nécessite pas d'infrastructure technique spécifique.

L'enregistrement des interactions entre le sujet et le logiciel testé est une autre possibilité. Un logiciel archive les actions sur le clavier, les positions du curseur à l'écran ou encore les contrôles (menus, boutons...) activés par le sujet pendant les tests. Des logiciels spécialisés sont disponibles pour collecter ce type d'information quelle que soit l'application informatique évaluée. Un avantage de cette méthode réside dans la mise à disposition de l'évaluateur des données complètes de l'interaction sujet – logiciel sous une forme facilement exploitable informatiquement.

Il est également possible d'inviter le sujet à remplir **un questionnaire**. A ce propos, nous rappelons que cette méthode pose des contraintes méthodologiques souvent négligées. En effet, la réalisation d'un questionnaire pertinent et efficace est loin d'être triviale. Par exemple, Dumas (2003) et Kirakowski (2000) expliquent toutes les difficultés à surmonter pour élaborer un questionnaire correct. En conséquence, de nombreux évaluateurs préfèrent utiliser des questionnaires standard élaborés par des spécialistes de la question, tels que QUIS (*Questionnaire for User Interaction Satisfaction*, 2007) ou SUMI (*Software Usability Measurement Inventory*, 2007).

La collecte d'informations par **interview** du sujet est encore une autre méthode. L'évaluateur discute avec le sujet sous forme non structurée ou en se référant à un certain cadre identifiant les questions à aborder. Cette méthode peut s'avérer riche d'enseignements mais elle exige de l'évaluateur une certaine constance dans les points abordés, dans son attitude vis-à-vis du sujet et dans la manière dont l'interview est transcrite.

Enfin, la méthode '**think aloud**' consiste à demander au sujet à parler à haute voix pendant qu'il évalue le logiciel. Les évaluateurs se basent sur ces commentaires pour émettre un avis sur le logiciel.

Naturellement, il est possible, voire recommandé, de combiner ces différentes techniques de collecte de données au sein d'une campagne globale d'évaluation.

2. Plan d'évaluation

Dans cette section, nous proposons une vue d'ensemble du plan d'évaluation des logiciels que nous avons conçus.

Pour commencer, il est nécessaire de préciser l'objet de l'évaluation. En d'autres termes, que souhaitons-nous évaluer ? Les contributions de nos travaux peuvent être potentiellement scindées en deux volets :

- les prototypes logiciels en tant qu'outils qui incluent des nouvelles techniques de visualisation d'information (telles que le concept d'ellimap pour visualiser des données hiérarchiques) et
- les prototypes logiciels utilisés comme outils de supervision et d'analyse d'une plateforme électronique de coopération.

Le premier élément appartient plutôt aux disciplines de la visualisation d'information et de l'interaction homme-machine. Le second point se situe dans l'étude des systèmes coopératifs et il fait notamment appel à des éléments relatifs à des disciplines telles que la sociologie ou le management. Néanmoins, il est évident que les deux aspects sont interreliés.

Au moment de la définition de notre politique en la matière, nous avons été confrontés au défi classique posé par une évaluation : l'élaboration, la mise en œuvre et l'analyse d'une campagne d'évaluation de logiciels est une entreprise coûteuse en temps et en ressources. Nous n'avons donc pas pu réaliser l'ensemble des études possibles sur nos propositions.

Comme nous l'avons déjà précisé précédemment, nous avons choisi de mettre l'accent sur la technique des ellimaps en tant que nouvelle approche pour visualiser des hiérarchies pondérées.

Tout au long de nos travaux de conception et de réalisation de logiciels de visualisation, nous avons utilisé une combinaison de méthodes d'évaluation formative et sommative. Une vue d'ensemble de notre démarche est proposée ci-dessous.

Table 24: Plan d'évaluation du prototype

Type d'évaluation	Méthode	Objet évalué	Echantillon
<i>Formative F1</i>	<i>Utilisation puis discussion ouverte</i>	<i>Modules 'ellimap' et 'clock view' du prototype (version 1)</i>	<i>2 superviseurs de plateformes collaboratives</i>
<i>Formative F2</i>	<i>Démonstration puis discussion ouverte</i>	<i>Prototype avec 'ellimap' et 'treemap' (version i)</i>	<i>2 superviseurs de plateformes collaboratives</i>
<i>Formative F3</i>	<i>Démonstration puis discussion ouverte</i>	<i>Modules 'ellimap' et prototype (version j)</i>	<i>3 informaticiens</i>
<i>Formative F4</i>	<i>Test empirique d'utilisabilité (diagnostic)</i>	<i>Module 'ellimap' du prototype (version 2)</i>	<i>10 utilisateurs potentiels (non informaticiens)</i>
<i>Sommative S1a</i>	<i>Test empirique d'utilisabilité (comparatif)</i>	<i>Comparaison des vues 'ellimap' et 'treemap' du prototype (version 3)</i>	<i>34 utilisateurs potentiels (dont 9 informaticiens)</i>
<i>Sommative S1b</i>	<i>Questionnaire</i>	<i>Satisfaction subjective de l'utilisateur du prototype en tant qu'outil générique de visualisation (version 3)</i>	<i>34 utilisateurs potentiels (dont 9 informaticiens)</i>

L'évaluation formative a été mise en œuvre en cours de développement des prototypes logiciels. Compte tenu des ressources limitées dont nous disposions, nous avons dû choisir les éléments spécifiques sur lesquels nous focaliserions l'évaluation. En toute logique, il nous a paru plus important d'évaluer les éléments novateurs car ils étaient plus susceptibles d'être sources de difficulté que les aspects plus classiques. Par conséquent, nous avons réalisé des tests d'évaluation formative focalisés sur les techniques de visualisation que nous avons imaginées, c.-à-d. les vues '*clock view*' et '*ellimap*'. Nos démarches d'évaluation formative sont détaillées dans la section 3 de ce chapitre.

En ce qui concerne l'évaluation sommative, nous avons identifié deux éléments à analyser spécifiquement. En premier lieu, nous avons estimé qu'il était nécessaire d'appréhender les éventuels apports de la technique '*ellimap*' par rapport à l'approche *treemap* plus classique. Dès lors, nous avons décidé de réaliser un test empirique comparatif basé sur des mesures au niveau de tâches particulières. Ensuite, il était naturel de procéder à une évaluation plus globale de l'ensemble du prototype. Dans ce but, nous avons conduit une évaluation basée sur un questionnaire pour mettre en évidence le niveau de satisfaction subjective d'utilisateurs du prototype. Les opérations d'évaluation sommative sont présentées dans la section 4 de ce chapitre.

Le positionnement temporel relatif des différents tests d'évaluation est présenté sur la Fig. 153.

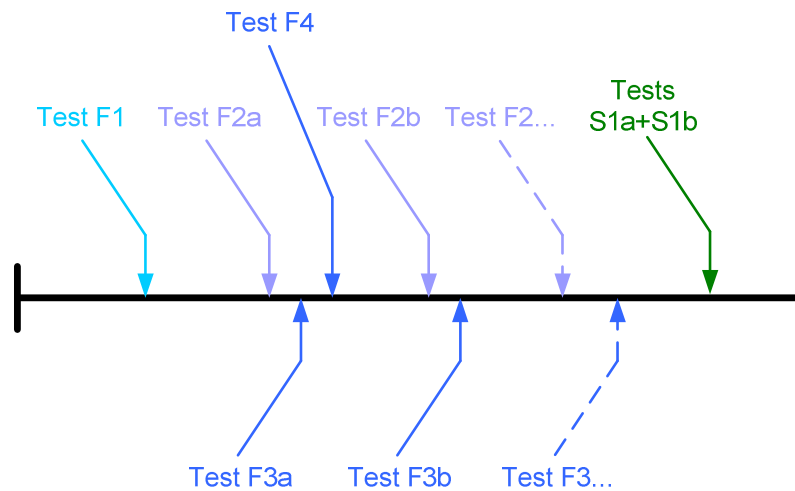


Fig. 153: Succession des évaluations

3. Evaluation formative

3.1 Test F1

3.1.1. Méthodologie

Le premier test d'évaluation formative fut réalisé sur une première version du prototype qui incluait la vue '*clock view*' et une version rudimentaire de l'ellimap. A ce moment, la vue ellimap n'existait que dans sa version initiale mais l'axe selon lequel la division de l'espace d'affichage était défini pouvait être configuré (*toujours vertical, toujours horizontal, alterné en commençant verticalement, alterné en commençant horizontalement*). De plus, la seule politique disponible de coloriage des ellipses était le coloriage alterné d'un niveau à l'autre. La stratégie de positionnement des étiquettes (*labels*) était très simple et il n'existait pas de possibilité de choix de la propriété affichée comme étiquette. Par contre, le couplage avec une vue en arbre classique était déjà intégré.

Cette première confrontation avec des utilisateurs avait pour but de collecter une première impression sur les nouvelles vues proposées. Deux superviseurs de plate-forme coopérative furent impliqués dans cette évaluation.

Au point de vue méthodologique, les deux sujets furent impliqués de manière individuelle. Le test complet a duré 30 minutes. L'évaluation s'est déroulée en trois parties.

La première partie avait pour objectif de collecter une impression sur le caractère intuitif de la '*clock view*' et de l'ellimap. Les deux sujets furent placés successivement devant chacune des vues sans aucune explication. Il leur était demandé d'interagir avec elles et de donner leurs commentaires à voix haute.

Dans la seconde partie, l'évaluateur a demandé aux sujets d'expliquer si les vues proposées semblaient présenter de l'intérêt pour les aider à superviser leur propre plate-forme coopérative.

Enfin, au cours de la troisième partie, il était demandé aux sujets s'ils voyaient du potentiel pour ces vues dans d'autres contextes que leur propre plate-forme.

3.1.2. Résultats

Nous discuterons séparément des résultats relatifs à chacune des deux vues.

En ce qui concerne la '*clock view*', les points positifs les plus notables mentionnés par les deux sujets sont les suivants.

- La métaphore de l'horloge est originale.
- Le codage du type d'interaction par le biais de couleurs est perçu positivement.
- La notion de vue égocentrique semble comprise.

Les faiblesses les plus importantes sont listées ci-dessous :

- Difficulté à comprendre les zones de texte associées aux informations détaillées sur les interactions.
- Problèmes d'interactivité avec la vue (p.ex. sélection difficile des rayons représentant les interactions).
- Problème de lisibilité de la vue (p.ex. menu contextuel affiché lors de clicks dans l'horloge pas assez intelligible).

Concernant la supervision de leur plate-forme, les sujets ont vu de l'intérêt pour représenter l'historique des activités et pour aider à planifier la promotion du site web. Ils ont perçu la vue comme un outil complémentaire de statistiques.

Dans une perspective plus large que leur propre plate-forme, un des sujets a précisé que la vue pourrait compléter utilement la suite logicielle Lotus Notes qu'il utilise. Le second sujet était

plus dubitatif mais a néanmoins mentionné que la vue pourrait éventuellement être utile dans le contexte de l'organisation personnelle de son emploi du temps.

Selon les sujets, la vue ellimap présente les avantages suivants :

- une compréhension rapide des symboliques ensemblistes,
- une perception immédiate de la structure hiérarchique,
- le couplage avec la vue en arbre,
- la possibilité de régler la profondeur d'affichage de la hiérarchie,
- la représentation du poids des nœuds par la surface des ellipses (pour un des sujets).

Par contre, elle présente également des faiblesses.

- Le réglage du facteur d'espacement entre les ellipses fut mal compris par un des sujets.
- La modification de présentation offerte par les options de paramétrage de l'orientation de division de l'espace d'affichage à chaque niveau hiérarchique fut jugée perturbante.

L'intérêt de l'ellimap tant dans pour le contexte de leur propre plate-forme que dans un cadre plus général fut jugé prometteur par les deux sujets. Son potentiel fut perçu comme supérieur à celui de la '*clock view*'. Les sujets mentionnèrent que la vue ellimap pourrait être utilisée comme vue d'ensemble d'un planning ou encore pour aider à gérer les livrables d'un projet. Un des deux sujets a émis la suggestion que l'utilisation de couleurs pourrait augmenter la valeur de l'ellimap.

Nous avons tiré deux grandes conclusions de ce premier test. Tout d'abord, la vue ellimap semble plus prometteuse que la vue '*clock view*'. Deuxièmement, la vue ellimap mérite d'être améliorée tant au niveau de sa définition conceptuelle que de son intégration dans le prototype. Par conséquent, nous avons décidé de focaliser la majeure partie de nos travaux de recherche sur l'ellimap.

3.2 Tests F2

3.2.1. Méthodologie

L'évaluation F2 fut réalisée sous la forme d'une série d'instances qui ont accompagné le développement du prototype. Le but était de confronter régulièrement l'état courant du prototype avec des utilisateurs potentiels.

Comme nous avons choisi d'axer une partie importante de nos travaux sur la vue ellimap, elle fut au centre des préoccupations des tests F2. Cependant, très rapidement, nous avons identifié l'intérêt scientifique d'inclure dans le prototype une vue *treemap* afin de pouvoir disposer d'une base de comparaison. Dès lors, la vue *treemap* fut également intégrée dans les tests F2.

Afin d'être représentatifs du contexte de l'utilisation planifiée pour le prototype, deux animateurs de plate-formes coopérative furent choisis. Il s'agit de personnes différentes que celles concernées par le test F1. Pour être précis, il faut noter que les deux premières instances d'évaluation ne concernaient qu'un seul sujet qui fut rejoint ensuite par un collègue.

L'évaluation était réalisée en deux étapes. Dans un premier temps, une démonstration était organisée par un évaluateur et les améliorations par rapport à la version précédente étaient mises en avant. Une manipulation avec le prototype fut également intégrée au test à partir de la troisième séance. Ensuite, dans une seconde étape, les sujets expliquaient sous forme d'une discussion non structurée les avantages et inconvénients de la version du prototype présentée. En moyenne, une séance durait de 45 minutes à une heure.

3.2.2. Résultats

Nous ne détaillerons pas ici l'ensemble des remarques énoncées au cours de ces séances d'évaluation. Globalement, les observations des sujets étaient de deux ordres : d'une part des questions relatives à la supervision de leur plate-forme coopérative, d'autre part des commentaires sur la technique de visualisation proprement dite.

En ce qui concerne l'utilisation du prototype pour superviser leur plate-forme, les sujets étaient essentiellement préoccupés par des questions liées aux types de données qu'ils pourraient visualiser et à l'intégration éventuelle du prototype dans leur plate-forme. Ces questions visaient plutôt les volets technologique et organisationnel. Bien qu'essentielles dans le cas d'une application mise en production, elles sont moins importantes dans la réflexion sur les techniques de visualisation proprement dites.

Dans l'optique d'une amélioration incrémentale de la vue ellimap, nous avons concentré nos ressources sur la prise en compte des remarques relatives à cette technique particulière et à son intégration efficace dans une application informatique stable et conviviale.

Afin de donner naissance à des commentaires de bonne qualité, nous avons choisi dès la seconde séance de visualiser des données réelles extraites de la plate-forme des deux sujets. Nous les mettions dès lors en situation quasi réelle et ils pouvaient plus facilement expliquer les avantages et les lacunes de la version courante du prototype. Cette utilisation d'un scénario ayant du sens pour les sujets nous est apparue essentielle pour collecter des informations pertinentes.

A ce niveau, les observations des sujets étaient pour la plupart très ponctuelles et destinées à être prises en compte pour les versions suivantes du prototype. A titre d'exemples, les remarques ont notamment porté sur le choix des couleurs utilisées par défaut pour colorer les ellipses ou sur les options à intégrer pour choisir l'étiquette (*label*) associée aux nœuds.

Dans un souci de concision, nous résumons les conclusions d'ordre plus général.

En premier lieu, les deux sujets furent très rapidement séduits par l'idée d'utiliser la propriété d'inclusion pour représenter des données hiérarchiques.

Ensuite, l'enrichissement progressif des stratégies de coloriage des surfaces représentant les nœuds fut fortement apprécié.

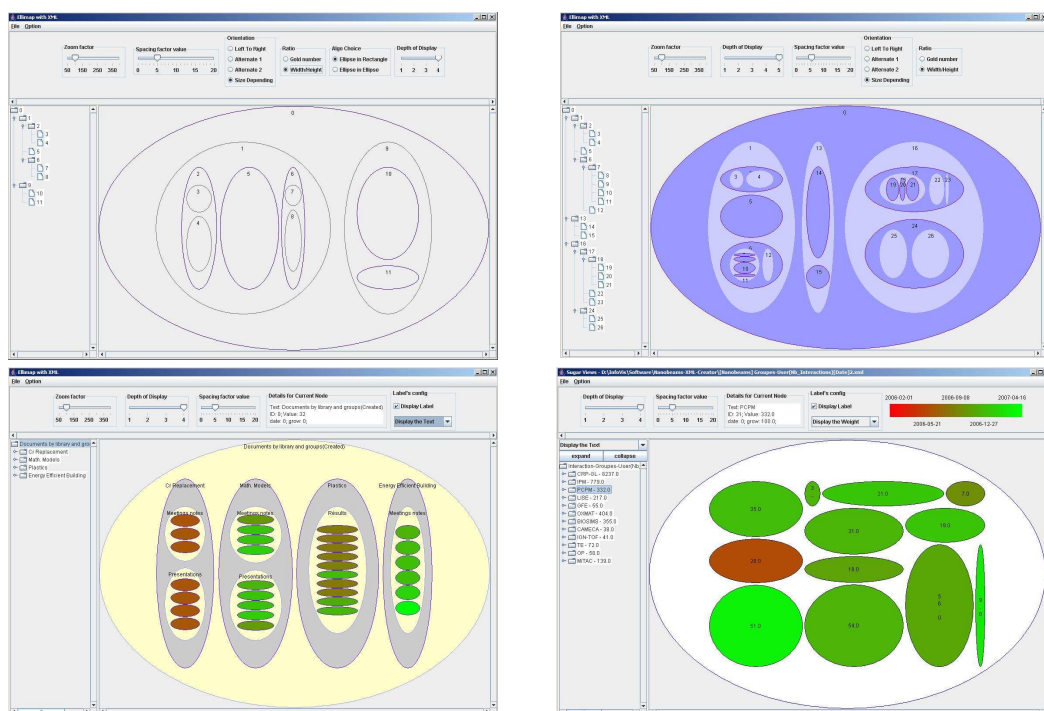


Fig. 154: Illustration des évolutions de la vue 'ellimap' au cours des tests formatifs

Les sujets évoquèrent aussi la question de la comparaison entre les vues *treemap* et *ellimap*. Selon eux, la vue *ellimap* met mieux en évidence la structure des données. Elle leur apparaît plus lisible. Pour être rigoureux, nous attirons cependant l'attention sur le fait que les exemples utilisés lors de ces discussions faisaient référence à des jeux de données de taille faible à moyenne, ce qui à notre avis, défavorise la vue *treemap* dont l'efficacité augmente avec le nombre de nœuds à visualiser. Les hiérarchies utilisées comme support des tests F2 comptaient de quelques dizaines à quelques centaines de nœuds.

A titre illustratif, les figures ci-dessous montrent l'évolution de l'interface de la vue *ellimap* du prototype pendant la période durant laquelle les tests F2 et F3 furent réalisés. Elles démontrent l'intérêt et l'apport que des tests formatifs répétés peuvent générer lors de la conception de techniques de visualisation et d'interfaces homme-ordinateur.

3.3 Tests F3i

3.3.1. Méthodologie

L'évaluation F3i fut également mise en œuvre sous la forme d'une série d'instances tout au long du développement du prototype. Ici encore, le but était de confronter régulièrement l'état courant du prototype avec des utilisateurs potentiels. Comme dans les tests F2, les vues *ellimap* et *treemap* furent concernées par les évaluations F3.

Néanmoins, la différence majeure avec la série de tests F2 concerne les sujets interrogés. Dans les tests F3, les personnes impliquées dans l'évaluation étaient des informaticiens. Il s'agissait de professionnels extérieurs à notre centre de recherche mais intéressés par le potentiel offert par les techniques de visualisation dans d'autres contextes que la supervision de plates-formes coopératives.

L'évaluation était réalisée en deux étapes. Dans un premier temps, une démonstration était organisée par un évaluateur et les améliorations par rapport à la version précédente étaient mises en avant. Une manipulation avec le prototype fut également intégrée au test à partir de la quatrième séance. Ensuite, dans une seconde étape, les sujets expliquaient sous forme d'une discussion non structurée les avantages et inconvénients de la version du prototype présentée. En moyenne, une séance durait de 45 minutes à une heure.

3.3.2. Résultats

Afin de placer les sujets en situation quasi réelle, nous avons discuté avec eux de jeux de données qui les concernaient et qui pouvaient être visualisés par notre prototype. Ils identifièrent trois exemples : la gestion des heures supplémentaires, la gestion des photocopieuses et des consommables associés et la gestion de données à caractère social. Nous avons donc retranscrit les données brutes qu'ils nous ont fournies pour les structurer sous la forme d'un arbre.

Les informaticiens étaient particulièrement attentifs à la genericité de notre prototype. De leur point de vue, le fait que notre outil soit découplé de l'application informatique qui gère les données visualisées représente un avantage majeur. En effet, ils ont adhéré à l'idée d'une utilisation possible par leur organisation d'une telle solution mais sous la condition de garder la main sur la production et la gestion des données par les applications existantes. De plus, les étapes de conception et de production des documents XML fournis en entrée à notre prototype devaient rester sous leur contrôle. Ces observations nous ont confortés dans notre choix au niveau de l'architecture logicielle que nous avons choisie.

De manière similaire aux tests F2, la confrontation du prototype avec des informaticiens professionnels nous a permis d'améliorer ses fonctionnalités. Parmi les diverses propositions d'amélioration, nous pouvons mentionner à titre illustratif :

- l'ajout d'une fonction pour mémoriser la configuration de la vue (c.-à-d. les valeurs des paramètres définissant par exemple la stratégie de coloriage ou de division récursive des espaces d'affichage),

- le besoin de disposer d'une interface utilisateur simplifiée (c.-à-d. avec des possibilités de configuration limitées) destinée à des personnes qui ne sont pas expertes face à un ordinateur.

Ces tests présentaient un double intérêt. D'une part, ils nous ont permis de faire évoluer le prototype. D'autre part, ils ont fourni de nouveaux exemples d'application des ellimaps.

3.4 Test F4

3.4.1. Méthodologie

Le test F4 constitue la première évaluation qui puisse être considérée comme un test empirique d'utilisabilité reposant sur une méthodologie plus formelle. En effet, les tests précédents se basaient essentiellement sur un nombre très réduit de sujets et prenaient la forme de discussions informelles non structurées.

Il est important de rappeler que d'un point de vue méthodologique, un test empirique d'utilisabilité n'a pas pour ambition de constituer une approche de type '*research study*'. Comme Dumas (2003) le précise, des principes tels qu'isoler une variable indépendante et disposer d'échantillons permettant le calcul de tests statistiques ne s'appliquent pas à un test de diagnostic (*diagnostic usability test*). Cela ne leur empêche pas d'être largement utilisés par les professionnels de l'utilisabilité et de fournir des éléments utiles à l'amélioration des produits. La seule contrainte intellectuelle consiste à être conscients de la portée de leurs résultats.

Nous avons réalisé une évaluation préliminaire du concept d'ellimap. Cette étude avait pour but premier de déterminer si des sujets peuvent comprendre sans explications que l'ellimap représente des données hiérarchiques. En complément, nous voulions aussi évaluer le caractère intuitif de l'interface utilisateur. Par exemple, le sujet découvre-t-il par lui-même le couplage entre la vue ellimap et la vue en arbre ? Afin d'éviter les biais qui pourraient résulter de la connaissance préliminaire d'un domaine applicatif particulier par les utilisateurs de test, nous avons décidé de visualiser un jeu de données complètement abstrait. Le fichier chargé dans le prototype est une structure arborescente avec des nœuds pondérés. La structure de données utilisée compte 26 nœuds répartis sur 5 niveaux de profondeur. (cf. Fig. 155). Nous sommes bien conscients qu'il s'agit d'un exemple de structure hiérarchique de taille réduite. Nous l'avons cependant choisi dans cette première étape de notre démarche car nous visions à évaluer en premier lieu la compréhension du concept même d'ellimap.

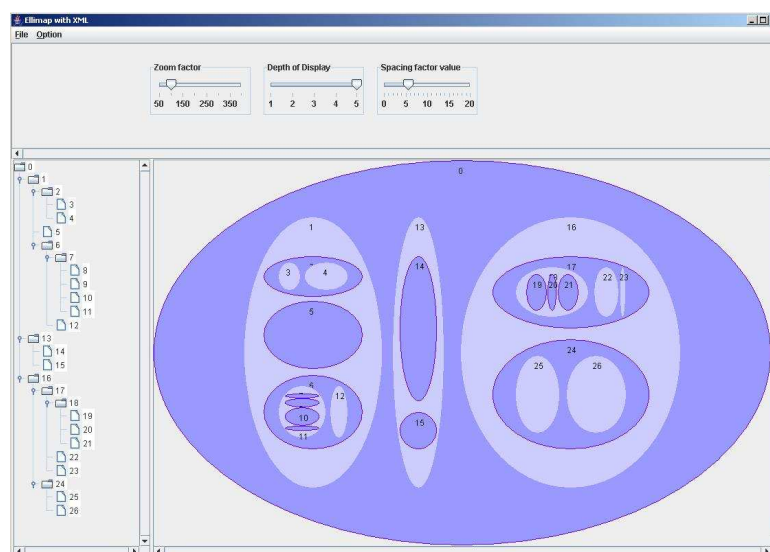


Fig. 155: Ellimap représentant le jeu de données utilisé pour le test F4

Chaque test concerne trois évaluateurs et un utilisateur. Le test est composé de trois parties : une exploration libre (EL), une démonstration (DM) et une seconde utilisation (SU). La durée totale du test est de 15 minutes. Au cours de la première partie du test (EL), le sujet est placé face à un ordinateur sur lequel le prototype ellimap est démarré. Il ne reçoit aucune explication, ni à propos du but du logiciel ni à propos de ses fonctionnalités. Un des évaluateurs lui demande d'explorer l'application à sa guise, de commenter les fonctionnalités qu'il découvre et d'expliquer ce que la visualisation représente selon lui. Cette partie dure 5 minutes. Pendant cette phase, les trois évaluateurs n'aident pas du tout le sujet. Ils transcrivent seulement ce qu'ils observent, de même que les commentaires du sujet. Le formulaire d'évaluation leur demande néanmoins de noter quatre observations formelles : si le sujet a compris que l'ellimap visualise une hiérarchie ; s'il a découvert le couplage entre la vue ellimap et la vue en arbre ; s'il a compris que la taille des ellipses est représentative du poids des nœuds et s'il a compris le mécanisme de sélection du niveau de profondeur affiché.

La seconde phase du test (DM) dure 3 minutes pendant lesquelles un des évaluateurs prend la main et réalise une démonstration du prototype. Les deux autres évaluateurs notent les commentaires du sujet. Ensuite, le sujet est invité à utiliser une seconde fois le logiciel pendant une période de 7 minutes (SU). Les évaluateurs lui demandent ce qu'il pense des différentes fonctionnalités, notamment en termes de facilité d'utilisation et de potentiel d'usage dans des contextes applicatifs. Cette partie du test est conduite sous la forme d'une discussion informelle. Le formulaire utilisé par les évaluateurs ne contient plus de directives sur des observations formelles à renseigner. Les évaluateurs transcrivent simplement les commentaires du sujet et son comportement vis-à-vis du logiciel.

L'échantillon contient dix individus (5 hommes – 5 femmes). Aucun d'entre eux n'est informaticien. Il s'agit de professionnels actifs dans les secteurs de l'administration d'une organisation (5 sujets), de la biologie (3 sujets) et de la physique (2 sujets). Ils sont tous familiers avec l'usage de l'ordinateur mais ne sont des experts ni en visualisation d'information, ni en informatique graphique (*computer graphics*).

3.4.2. Résultats

La taille réduite de l'échantillon ne permet pas de tirer des conclusions statistiquement significatives. Néanmoins, la méthode de test utilisée (*discount usability testing*, Shneiderman et Plaisant, 2005, p. 148) fournit une évaluation relativement correcte de la perception initiale du prototype par des utilisateurs réels. Une telle approche est souvent considérée comme efficace pour mettre en évidence les avantages principaux et les défauts majeurs d'une interface utilisateur.

Notre commentaire se focalisera sur trois types de résultats : des données quantitatives sur la reconnaissance par le sujet de certaines fonctionnalités de base du logiciel (cf. partie EL du test), un résumé des commentaires formulés par les utilisateurs de test (cf. parties DM et SU du test) et des exemples de champs d'application potentiels qu'ils ont mentionnés (cf. partie SU du test).

En premier lieu, il est remarquable que tous les sujets aient trouvé en moins de deux minutes et sans aucune explication que l'ellimap représente des données hiérarchiques et que la vue en arbre est couplée à la vue ellimap. Huit sujets sur dix ont découvert le but du curseur (*slider*) qui contrôle la profondeur d'affichage pendant la première partie du test. Enfin, seulement deux d'entre eux ont explicitement mentionné de leur propre initiative que la taille de l'ellipse est un indicateur de l'importance des nœuds. Sur cette base, nous pouvons tirer deux conclusions préliminaires. D'une part, l'ellimap semble être une visualisation intuitive de données hiérarchiques. D'autre part, le couplage des vues 'arbre' et ellimap est un élément essentiel qui aide l'utilisateur à découvrir le but de l'ellimap par lui-même. En effet, la plupart de sujets ont commencé l'exploration du prototype avec la vue en arbre, c.-à-d. avec une représentation graphique qu'ils connaissaient auparavant.

Deuxièmement, quatre sujets ont explicitement précisé de leur propre initiative (pendant les parties DM et SU du test) que l'ellimap est plus conviviale que la vue en arbre. Trois

utilisateurs ont expliqué qu'ils préféraient des figures géométriques curvilignes à des formes ayant des angles vifs. Parmi les autres remarques positives, on peut encore relever :

- *'l'ellimap aide à voir les choses d'une manière différente' ;*
- *'l'ellimap représente bien les proportions entre les éléments d'un même niveau' ;*
- *'l'ellimap montre la profondeur des choses'.*

La perception globale du prototype s'est avérée clairement positive. Certains sujets ont cependant pointé des limitations du prototype. Nous avons notamment noté des demandes pour pouvoir annoter les ellipses, pour pouvoir les déplacer et pour disposer de possibilités plus avancées pour les colorier. Un des sujets a aussi pris conscience du fait que les ellipses inscrites ne remplissaient pas toute la surface de l'ellipse. D'un point de vue général, la conclusion est que l'ellimap semble être une représentation graphique dont la compréhension est assez facile. Néanmoins, le logiciel présente encore des possibilités d'amélioration au niveau de ses fonctionnalités.

Troisièmement, les dix sujets ont identifié en moins de cinq minutes toute une série d'applications potentielles pour le concept d'ellimap. La gamme des possibilités apparaît assez large ainsi que l'illustrent les exemples suivants : hiérarchies de documents, organigrammes d'entreprise, thésaurus, arbres phylogénétiques. Nous fûmes assez surpris de constater qu'aucun des utilisateurs de test n'a jugé inutile cette nouvelle représentation alors qu'ils ne disposaient que de quelques minutes pour se familiariser avec une approche qui leur était totalement inconnue.

D'un point de vue méthodologique, nous sommes raisonnablement confiants que les sujets n'aient senti aucune pression qui les incite à formuler des avis particulièrement favorables, en particulier parce que les évaluateurs n'avaient aucune relation hiérarchique avec eux.

4. Evaluation sommative

4.1 Méthodologie

Avant tout, il peut être utile de rappeler une différence fondamentale entre un test d'utilisabilité et une expérience contrôlée. Shneiderman et Plaisant (2005, p. 144) expliquent que le résultat du premier type de test est un rapport qui regroupe des propositions d'amélioration alors que le résultat du second type de test consiste en la validation ou le rejet d'une hypothèse. Nous avons conçu notre campagne globale d'évaluation en tenant compte de cette différence dans les objectifs des différentes méthodes de test.

D'un point de vue général, notre plan d'expérience se base sur le tutorial d'application des statistiques à l'étude de l'interaction homme-machine de Noirhomme (2005), sur le tutorial d'Andrews (2007) relatif à l'évaluation des techniques de visualisation, sur les manuels de statistiques de Dagnelie (1998, 2006), Howell (1998) et Saporta (2006) ainsi que sur les recommandations de Lewis (2006a, pp. 1285-1290).

4.1.1. Vue d'ensemble

La première étape lors de la conception d'un plan d'expérience consiste à identifier le but de l'exercice. Dans notre cas, il s'agit de réaliser une évaluation comparative de deux techniques de visualisation : les *treemaps* et les ellimaps. Nous ne visons donc pas à mettre en œuvre à ce stade un test de diagnostic, à vocation formative. Des tests de ce type ont été réalisés antérieurement et relatés dans des parties précédentes de ce document.

Nous souhaitons comparer les *treemaps* et les ellimaps sur base de leur capacité respective à faire percevoir à l'utilisateur la structure du jeu de données et le poids relatif des nœuds. Nous avons choisi ces éléments car ils représentent les points forts de ces deux techniques.

On peut noter dans ce contexte que comparer la perception de la structure hiérarchique entre une technique de visualisation basée sur l'inclusion de formes géométriques (cf. ellimap ou *treemap*) et une visualisation classique en arbre aurait présenté moins d'intérêt car tant la théorie de la '*Gestalt*' que des travaux expérimentaux antérieurs ont montré les qualités de la vue en arbre pour ce type de tâche. Par contre, comme cela a été expliqué précédemment (cf. p. 180), la vue en arbre présente par contre d'autres faiblesses, notamment au point de vue de l'occupation de l'espace d'affichage.

Les hypothèses que nous testerons sont donc les suivantes :

- H1 : L'exactitude de la perception de la structure d'une hiérarchie diffère entre la représentation par *treemap* et par ellimap.
- H2 : Le temps nécessaire pour percevoir correctement la structure d'une hiérarchie est différent dans la représentation par *treemap* et par ellimap.
- H3 : L'exactitude de la perception du poids des nœuds d'une hiérarchie diffère entre la représentation par *treemap* et par ellimap.
- H4 : Le temps nécessaire pour percevoir correctement le poids des nœuds d'une hiérarchie est différent dans la représentation par *treemap* et par ellimap.
- H5 : La qualité de perception combinée de la structure et du poids des nœuds d'une hiérarchie est différente entre la représentation par *treemap* et par ellimap.
- H6 : Le temps nécessaire à une perception combinée correcte de la structure et du poids des nœuds d'une hiérarchie est différent entre la représentation par *treemap* et par ellimap.

Notre objectif est donc de réaliser une évaluation sommative qui nous permettra de formuler un avis complémentaire sur la qualité de la nouvelle technique que nous avons imaginée (ellimaps). Dans cette optique, il apparaît logique que nous ne visions pas à ce niveau la découverte de problèmes d'utilisabilité mais plutôt la collecte de mesures associées à des tâches.

Au cours d'un séminaire récent consacré à l'évaluation des techniques de visualisation, Keith Andrews (2006) a mis en évidence quatre défis importants qu'il considère, sur base de son expérience, comme les plus critiques dans ce contexte.

- Comme les prototypes implémentant de nouvelles visualisations sont souvent imparfaits (bogues, manque de fonctionnalités), il n'est pas équitable de les comparer à des logiciels éprouvés par l'expérience.
- Très souvent, les utilisateurs de test ne sont pas des utilisateurs réels mais plutôt des étudiants (en informatique).
- Il est extrêmement difficile de surmonter le biais lié à la familiarité des utilisateurs avec des interfaces traditionnelles. On constate souvent que la technique de visualisation connue 'gagne' dans un test comparatif qui la confronte avec une nouvelle approche.
- On observe régulièrement que l'évaluation concerne de 'mauvaises' tâches, comme par exemple des tâches qui ne sont pas représentatives ou des tâches trop simples.

Au cours de notre réflexion méthodologique, nous avons essayé de tenir compte de ces difficultés pour concevoir une évaluation la plus pertinente possible.

Tout d'abord, les deux techniques de visualisation évaluées ont été implémentées dans le même logiciel (cf. p. 274). Celui-ci présente exactement les mêmes fonctionnalités pour les *treemaps* et pour les ellimaps. Par exemple, le système d'affichage des *tooltips* ou la navigation dans le jeu de données sont identiques. Nous évitons ainsi les biais liés aux différences de maturité d'implémentation entre les logiciels évalués.

Nous discuterons ultérieurement (p. 254 et 264) du choix des utilisateurs de test mais nous pouvons déjà préciser à ce stade qu'il ne s'agira pas d'une classe d'étudiants en informatique ou d'élèves d'un cours de visualisation.

Troisièmement, nous pensons éviter le biais en faveur de la technique connue au préalable par le fait que les deux techniques évaluées sont inconnues pour la grande majorité des sujets du test. Nous collecterons par ailleurs de l'information relative à cette connaissance préalable éventuelle pour commenter le plus correctement possible nos résultats expérimentaux.

Nous discuterons des tâches évaluées dans la section 4.1.4.a (p. 255) mais nous pouvons déjà affirmer à ce stade que nous nous sommes efforcés de considérer des tâches représentatives de ce que nous voulions évaluer : la perception de la structure hiérarchique et du poids des nœuds.

Enfin, nous discuterons dans la section 4.1.5 du plan d'expérience que nous avons mis en œuvre. C'est à ce niveau que seront définies les variables indépendantes et les variables dépendantes de notre campagne d'évaluation. Nous y expliquerons également comment nous avons combiné les deux types de variables pour élaborer notre méthodologie.

4.1.2. Techniques de visualisation évaluées

Nous avons indiqué dans notre revue de l'état de l'art que la technique de visualisation par *treemap* a fait l'objet de plusieurs adaptations depuis la proposition initiale de Shneiderman. Par ailleurs, nous décrivons dans ce document plusieurs évolutions de la technique des ellimaps. Bien que cela présente certainement un intérêt indéniable, nous ne disposons pas des ressources nécessaires pour réaliser des évaluations comparatives de toutes les variantes de ces deux techniques. Par conséquent, nous avons limité cette campagne d'évaluation formelle à une seule variante de chacune des deux techniques.

En ce qui concerne les *treemaps*, nous avons choisi la variante '*squarified treemap*' car elle est souvent présentée comme l'une des variantes les plus abouties. Au niveau des ellimaps, nous avons retenu la variante basée sur l'algorithme à division dépendante (cf. p. 195). Au moment où l'évaluation sommative a été réalisée, il s'agissait de la version proposant le meilleur compromis parmi celles dont nous disposions. Il faut noter que l'algorithme DGC (cf. p. 202) n'avait pas encore été conçu à cette époque.

4.1.3. Population

Nous avons identifié les caractéristiques de la population à laquelle s’adressent les techniques de visualisation que nous devons comparer. En fait, plus précisément, ces propriétés définissent la population à laquelle s’appliquent les différences entre les ellimaps et les *treemaps* qui pourraient éventuellement être mises en évidence par nos tests statistiques.

Les variables retenues pour décrire les membres de la population étudiée sont les suivantes :

- Niveau d’expérience
 - avec l’informatique et les ordinateurs
 - avec des techniques de visualisation d’information
 - avec des logiciels de visualisation d’information
 - avec les *treemaps*
 - avec les ellimaps
- Education
 - niveau
 - discipline
- Profession
- Age
- Genre

Nous expliquons ci-dessous pourquoi ces variables sont importantes pour notre évaluation.

Le **niveau d’expérience** est un premier élément crucial.

Nous considérons tout d’abord que les utilisateurs doivent posséder une certaine connaissance de l’informatique et des ordinateurs. Nous ne cherchons pas à limiter la population étudiée aux informaticiens mais les personnes évaluées doivent néanmoins disposer d’une certaine habitude avec les outils informatiques. Dans le cas contraire, les biais seraient tels que nous ne pourrions estimer dans quelle mesure les résultats sont dus au manque de familiarité avec l’informatique ou au logiciel testé.

Ensuite, comme nous évaluons des techniques de visualisation, il est important de préciser si nous nous adressons à des personnes possédant déjà une expertise dans ce domaine. Nous avons identifié des éléments qu’il est utile de connaître à ce niveau.

Tout d’abord, comme nous ne souhaitons pas limiter la portée de notre évaluation aux experts en visualisation, il est intéressant d’appréhender le niveau de connaissances des techniques pour visualiser de l’information (p.ex. graphe de Gantt, graphe de Pert...) dont disposent les sujets des tests. Cet élément nous permettra de savoir si l’échantillon inclut des personnes qui n’ont aucune connaissance particulière dans ce domaine. Rappelons, à titre d’exemple, que la population visée ne se réduit pas à des étudiants ayant suivi un cours de visualisation d’information.

De manière complémentaire, il est important de connaître l’expérience éventuelle que les personnes possèdent en tant qu’utilisateurs d’un logiciel qui permet de visualiser des données.

Enfin, nous ne souhaitons pas limiter la population aux personnes disposant d’une connaissance préalable des techniques de type ‘*treemaps*’. En effet, une telle approche induirait un biais potentiel en faveur de ces techniques connues par rapport à la nouvelle approche des ellimaps (cf. biais cités par Andrews).

En second lieu, l’**éducation** des personnes constitue à nos yeux une variable importante pour caractériser le public cible. Cet élément sera décomposé en deux volets : le niveau d’éducation et la discipline de formation.

Le niveau d’éducation nous apparaît important car il peut influencer l’aptitude préalable des personnes à utiliser des représentations conceptuelles ou abstraites de la réalité. Comme les techniques de visualisation appartiennent à ce type de représentation, cet élément peut s’avérer être un facteur influençant les résultats. La population que nous visons regroupe les

personnes ayant accompli avec succès un cycle de formation de l'enseignement secondaire supérieur (c.-à-d. niveau Bac en France) car elles sont les plus susceptibles de devenir des utilisateurs des deux techniques de visualisation concernées par les tests.

Dans le même ordre d'idées, la discipline dans laquelle les personnes ont reçu une formation peut influencer le test. Nous ne souhaitons pas réduire sur cette base notre évaluation à une catégorie de personnes. En particulier, nous ne limitons pas notre évaluation aux diplômés ou étudiants en informatique. Une telle restriction est en effet régulièrement observée dans les études expérimentales dans le domaine des interfaces homme-ordinateur (cf. p.ex. réflexion critique d'Andrews, 2006). Elle devient un biais si la population visée s'étend au-delà de cette catégorie de personnes.

Troisièmement, la **profession** exercée par une personne peut également jouer un rôle dans sa capacité à utiliser des techniques de visualisation. Ce critère peut renforcer ou limiter le biais relatif à la formation. En effet, il n'est pas exceptionnel qu'une personne disposant d'une formation littéraire exerce une profession à caractère plus mathématique (p.ex. analyste financier). Notre public cible inclura donc une gamme assez large de professions.

L'**âge** est le quatrième type de facteur que nous considérons. Notre prototype vise une population adulte classique. Nous souhaitons en effet éviter les problèmes spécifiques liés aux enfants et aux personnes âgées.

Pour terminer, la dernière variable retenue est le **genre** des personnes. Nous ne pensons pas que cet élément influence les résultats de manière significative mais nous l'avons néanmoins retenu par souci d'équité et de complétude des informations sur les tests.

4.1.4. Choix des observations

Les observations que nous considérerons seront essentiellement des mesures associées à certaines tâches que les sujets devront réaliser. Il nous faut donc identifier ces tâches et discuter des mesures que nous y associerons.

Nous ajouterons aussi une observation plus globale qui aura pour objectif de nous donner une indication sur la perception des deux techniques de visualisation par les utilisateurs de test. Cette évaluation subjective sera détaillée dans la section relative à la mise en œuvre pratique des tests (cf. 4.2.2.c, p. 271).

a. *Tâches*

Afin de préciser les tâches que nous inclurons dans le test et qui serviront de base à notre évaluation, nous avons exploré la littérature pour disposer d'exemples de travaux comparables. Nous avons uniquement pris en compte des évaluations présentant un lien clair avec notre problématique (p.ex. évaluation de *treemaps*, évaluation comparative de représentations de données hiérarchiques). Comme le précise Andrews (2007), il existe très peu d'évaluations formelles de visualisations de données hiérarchiques.

Les résultats de cette investigation sont présentés ci-dessous.

O'Donnel et al (2006) ont comparé les représentations '*PieTree*' et '*Tree*' pour visualiser des données arborescentes. Le jeu de données simulait des informations économiques sur une entreprise fictive. Leur étude expérimentale s'est basée sur les tâches suivantes :

- *trouver le nœud ayant la plus faible pondération,*
- *trouver le nœud ayant la plus grande pondération,*
- *trouver un nœud spécifique.*

Lee et al. (2006) ont comparé une nouvelle technique pour visualiser des graphes sous forme de hiérarchie (appelée '*TreePlus*') avec une technique existante de représentation de graphe ('*GraphPlus*'). L'exemple choisi faisait référence à des données sur des communications inter-personnelles.

Ils ont utilisé les tâches suivantes :

- *trouver un nœud spécifique,*
- *suivre un chemin dans le graphe,*
- *parmi les voisins d'un nœud clairement identifié, trouver ceux qui possèdent une caractéristique donnée,*
- *compter le nombre de nœuds qui possèdent une caractéristique donnée et qui sont situés à une distance '2' d'un nœud clairement identifié,*
- *trouver les nœuds qui sont les voisins directs de deux autres nœuds clairement identifiés,*
- *trouver le nœud qui possède le plus de voisins.*

Demian et Fruchter (2006a, 2006b) ont évalué le système 'CoMem' qui combine différentes représentations de données hiérarchiques. Le contexte de l'étude concernait un système informatisé de gestion d'informations relatives à des projets de construction. Les tâches retenues pour l'évaluation étaient les suivantes :

- *chercher un nœud spécifique,*
- *explorer l'ensemble des données sans idée très précise de ce que l'on cherche mais néanmoins en vue de trouver de l'information relative à une question générale (exploration).*

Irani et al. (2006) ont spécifiquement étudié la perception de la structure de données hiérarchiques en comparant les techniques 'Treemap' et 'CushionMap'. Ils utilisèrent l'exemple d'une structure de stockage de données informatiques. Leur évaluation s'est basée sur des tâches associées à la structure des données :

- *compter le nombre de répertoires dans la hiérarchie,*
- *trouver le répertoire avec le plus grand nombre de fichiers,*
- *compter le nombre de sous-répertoires dans un répertoire donné,*
- *compter le nombre de fichiers dans un sous-répertoire donné,*
- *trouver le répertoire contenant le plus de fichiers bitmap (.bmp),*
- *compter le nombre de sous-répertoires qui contiennent des fichiers bitmap.*

et sur d'autres reposant sur des comparaisons de taille :

- *trouver le plus petit répertoire de la hiérarchie,*
- *trouver le plus grand fichier dans la hiérarchie,*
- *trouver le plus grand fichier dans un répertoire donné,*
- *trouver le plus gros fichier 'mp3' dans la hiérarchie.*

Andrews (2007) a évalué quatre techniques de visualisation de données hiérarchiques : la vue en arbre (*tree view*), la pyramide d'information (*information pyramids*), le *treemap* et le navigateur hyperbolique (*hyperbolic browser*). Il s'est focalisé sur l'évaluation de tâches associées à la structure et à la topologie de la hiérarchie. Il exclut explicitement les tâches relatives à des attributs individuels des nœuds. Le jeu de données utilisé est une arborescence de fichiers informatiques.

Andrews utilise huit tâches différentes.

Deux tâches sont relatives à la vue d'ensemble de la structure (*overview*) :

- *trouver le sous-répertoire le plus profond dans un répertoire donné,*
- *au sein d'un répertoire donné, trouver le répertoire ayant le plus de sous-répertoires enfants.*

Il inclut aussi deux tâches de recherche (*search*) :

- *trouver le répertoire nommé R,*
- *trouver le fichier nommé F.*

Deux tâches consistent en une opération de comptage :

- *compter le nombre de sous-répertoires enfants d'un répertoire donné,*
- *compter le nombre de fichiers contenus dans un répertoire donné.*

Enfin, les deux dernières tâches sont des tâches de comparaison :

- *lequel des répertoires A et B possède le plus de sous-répertoires enfants ?*
- *lequel des répertoires A et B inclut le plus de fichiers ?*

Les exemples cités ci-dessus nous ont orientés pour choisir les tâches qui seront utilisées dans notre évaluation. A ce sujet, nous constatons que les tâches citées précédemment peuvent être regroupées en quatre catégories plus génériques. On distingue en effet des tâches :

- de **d'exploration** dans lesquelles l'utilisateur parcourt l'ensemble des données sans connaître a priori l'élément recherché mais plutôt dans le but d'être inspiré par certains d'entre eux,
- de **recherche par valeur** dans lesquelles l'utilisateur connaît exactement la valeur d'une propriété de l'élément recherché et cherche à le localiser dans l'ensemble des données,
- de **recherche par comparaison** dans lesquelles l'utilisateur se base sur un critère comparatif portant sur une propriété d'un élément pour identifier celui-ci et le localiser dans l'ensemble des données,
- de **perception de la structure** dans lesquelles l'utilisateur doit donner des informations sur l'organisation de l'ensemble des données.

La Table 25 fournit des exemples de questions concrètes associées à chaque type de tâche.

Table 25: Exemples de tâches pour les différents types

Type de tâche	Exemples
Exploration	Trouver des idées pour résoudre un problème donné en étant inspiré par un ensemble de données qui est parcouru librement (<i>p.ex. parcours d'un ensemble de photos pour imaginer la décoration d'une pièce</i>)
Recherche par valeur	Trouver un nœud dans l'ensemble des données en connaissant son identifiant (<i>p.ex. le nœud correspondant à M. Dupont</i>)
	Trouver les nœuds ayant une valeur connue pour une de leurs propriétés (<i>p.ex. fichiers de type '.bmp'</i>)
Recherche par comparaison	Trouver le nœud ayant la pondération la plus grande (<i>p.ex. le plus gros fichier en taille</i>)
Perception de la structure	Compter le nombre de nœuds enfants d'un nœud donné (<i>p.ex. nombre de fichiers contenus dans un répertoire</i>)
	Compter le nombre de niveaux hiérarchiques inférieurs à un nœud donné (<i>p.ex. nombre de niveaux hiérarchiques dans un département X</i>)

Avant de préciser quels types de tâches nous avons choisis, il peut s'avérer utile de se remémorer le contexte de notre démarche d'évaluation. Nous visons à comparer deux techniques : les *treemaps* et les ellimaps. Ces deux techniques permettent de visualiser la structure d'un jeu de données hiérarchiques ainsi que la pondération affectée à chaque nœud. Par conséquent, il nous a semblé logique de focaliser les tests sur des tâches de perception de la structure et de recherche par comparaison. Il s'agit en effet des points forts des deux techniques et donc des éléments par rapport auxquels il est important de déterminer d'éventuelles différences.

Le choix de tâches expérimentales pertinentes présente une difficulté non négligeable car il est important qu'elles soient représentatives des éléments que l'on souhaite évaluer. Bien qu'apparemment triviale, cette contrainte n'est pas aussi simple à satisfaire en pratique. Nous illustrerons cette difficulté avec les deux exemples de tâches que nous avons choisis pour notre propre évaluation.

Premièrement, nous cherchons à évaluer la perception de la structure hiérarchique des données. Si nous reprenons les exemples utilisés par Irani et al. (2006), nous observons que certaines tâches se focalisent uniquement sur la perception de la structure (p.ex. '*compter le nombre de répertoires dans la hiérarchie*'). Par contre, d'autres, que nous qualifierons d'hybrides, se décomposent en plusieurs sous-tâches. Par exemple, il s'agit de localiser un nœud donné N puis d'étudier un élément relatif à la structure des données à partir de ce nœud (p.ex. '*compter le nombre de fichiers dans un sous-répertoire donné*'). D'un point de vue rigoureux, si seule la perception de la structure est à évaluer, la tâche qui y est relative doit être isolée. Par conséquent, il faut supprimer la sous-tâche de localisation du nœud de référence N qui constitue en fait une tâche de recherche par valeur. D'un point de vue pratique, il est très difficile de supprimer complètement cette sous-tâche. Une piste pourrait consister à réduire son influence en aidant fortement l'utilisateur de test à la réaliser. Concrètement, cette approche pourrait se matérialiser en pré-sélectionnant le nœud N dans le jeu de données présenté à l'utilisateur (via des stratagèmes tels qu'y positionner initialement le curseur). Grâce à une telle opération, on pourrait espérer réduire fortement la difficulté pour réaliser la sous-tâche de localisation (recherche par valeur) de manière telle que la tâche globale (mesurée en temps et en taux de réussite) soit essentiellement influencée par la perception de la structure du jeu de données. Cette solution repose néanmoins sur l'hypothèse que la pré-sélection du nœud N rend immédiate sa localisation.

Deuxièmement, nous visons à évaluer la recherche par comparaison des poids affectés aux nœuds de la hiérarchie. Dans ce cas aussi, les tâches utilisées par Irani et al (2006) peuvent être décomposées de manière différente. On rencontre tout d'abord des tâches qui consistent uniquement en une recherche par comparaison (p.ex. '*trouver le plus grand fichier dans la hiérarchie*'). A côté de ces tâches 'pures', il existe aussi des tâches hybrides. La recherche du plus grand fichier dans un répertoire donné se décompose en une sous-tâche de localisation par valeur et une sous-tâche de recherche par comparaison.

La recherche du plus gros fichier 'mp3' est également une tâche hybride mais de nature plus complexe que la précédente. En effet, deux critères de filtre doivent être appliqués pour trouver le nœud recherché. Ils font référence à deux types de tâches : une recherche par valeur ('fichier mp3') et une recherche par comparaison ('le plus gros fichier'). Les interactions entre les deux types de recherche sont probablement complexes. Le sujet applique-t-il mentalement les filtres de manière séquentielle (et dans ce cas dans quel ordre ?) ou en parallèle ? Tous les sujets procèdent-ils d'une façon identique ? Ces questions nous incitent à la prudence.

En pratique, la plupart des tâches effectuées par les utilisateurs peuvent être qualifiées d'hybrides. Dès lors, afin de satisfaire à la contrainte de représentativité des tâches évaluées, nous avons choisi de considérer de telles tâches dans nos évaluations. Cependant, nous nous distinguerons des travaux antérieurs précités en identifiant comme telles ces tâches hybrides.

Dans le contexte de notre évaluation, nous définissons les tâches hybrides comme des combinaisons de tâches de perception de la structure des données, de recherche par comparaison et/ou de recherche par valeur.

Une autre contrainte s'impose à nous dans le choix des tâches à inclure dans notre évaluation. Afin d'observer d'éventuelles différences entre les techniques évaluées, nous devons choisir des tâches présentant une difficulté suffisante. En effet, le choix de tâches triviales risquerait de conduire soit à l'absence de différences entre les techniques, soit à des différences réelles mais non perceptibles par les moyens d'observation dont nous disposons. De plus, la plupart des tâches triviales présentent l'inconvénient d'un manque de représentativité de la réalité.

Trois catégories de tâches hybrides ont été retenues pour notre évaluation sommative :

- les tâches hybrides 'Valeur – Structure' (HyVS), combinant des opérations de recherche par valeur et de perception de la structure ;
- Tâches hybrides 'Valeur – Comparaison' (HyVC), combinant des opérations de recherche par valeur et de recherche par comparaison ;
- Tâches hybrides 'Valeur – Structure – Comparaison' (HyVSC), combinant des opérations de recherche par valeur, de perception de la structure et de recherche par comparaison.

Sur base de ces réflexions, les tâches listées dans la Table 26 ont été retenues pour les tests. Trois tâches ont été définies pour chaque catégorie.

Table 26: Tâches abstraites retenues pour les tests sommatifs

Type de tâche	Tâche
HyVS	
	<i>Identifier la suite des parents et ancêtres du nœud N</i>
	<i>Parmi les enfants du nœud N, identifier celui ayant le plus d'enfants</i>
	<i>Compter le nombre de nœuds frères du nœud N</i>
HyVC	
	<i>Classer n nœuds identifiés en ordre décroissant de pondération</i>
	<i>Trouver le nœud ayant la pondération la plus faible parmi n nœuds identifiés appartenant au même niveau hiérarchique</i>
	<i>Trouver le nœud ayant la pondération la plus forte parmi n nœuds identifiés n'appartenant pas au même niveau hiérarchique</i>
HyVSC	
	<i>Trouver le nœud ayant la pondération la plus forte parmi les parents de n nœuds identifiés</i>
	<i>Trouver le nœud ayant la pondération la plus faible parmi les enfants de n nœuds identifiés</i>
	<i>Parmi les feuilles descendant du nœud N, trouver celle ayant la pondération la plus forte</i>

b. Scénario

Lors d'une évaluation, situer les tâches à accomplir dans un scénario peut aider les sujets de tests à s'imaginer être dans un contexte réel d'utilisation du logiciel testé. Nous avons donc décidé d'intégrer dans un scénario les questions abstraites présentées dans la Table 26. Cette approche n'est cependant pas dénuée de tout danger car le choix du scénario doit éviter de favoriser certains sujets par rapport aux autres.

Nous avons choisi de baser notre scénario sur un exemple classique et universel de la vie courante, un exemple facile à comprendre et qui ne présuppose pas de connaissances spécifiques. Nous avons opté pour l'exemple de la vie d'un ménage. Ce scénario ne fait pas appel à une expertise dans une discipline scientifique ou à la connaissance préalable d'une technique particulière. D'une manière ou d'une autre, chacun est amené à devoir acheter des produits dans un magasin et à gérer⁴³ son budget personnel.

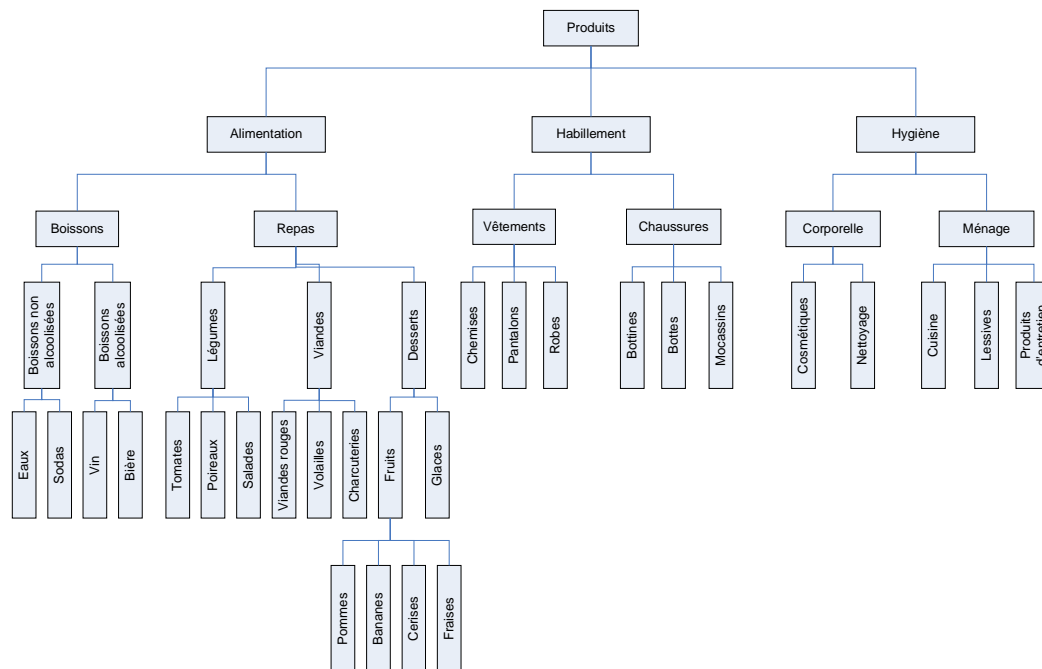


Fig. 156: Hiérarchie illustrant une classification de produits

Dans notre scénario, la hiérarchie fera donc référence à une classification d'objets de la vie courante qui nous appellerons des produits. Le poids des nœuds sera matérialisé par un budget attribué à chaque catégorie de produits. Les poids seront fixés aléatoirement pour les feuilles de la hiérarchie et seront calculés par sommation des poids des nœuds enfants pour les nœuds de niveau supérieur. La Fig. 156 illustre un jeu de données qui pourrait être utilisé dans les tests. Pour des raisons d'espace disponible et de lisibilité, nous n'avons pas représenté à ce stade une hiérarchie très grande ou très complexe. Cependant, les jeux de données utilisés en pratique peuvent être plus volumineux.

Afin de réduire le biais éventuel résultant de la connaissance préalable du prix des objets, les prix ont été délibérément fixés de manière aléatoire et ne sont plus conformes à la réalité. Les sujets en ont été informés et il leur a été précisé que les prix étaient exprimés dans une devise qu'ils ne connaissent pas et sont relatifs à un pays imaginaire. Pour crédibiliser ce scénario, il leur a été expliqué que, par exemple, le prix de la viande sur une île sans mammifère était prohibitif alors que le prix du homard dans un village côtier envahi par ces animaux pouvait

⁴³ Il est notoire que la rigueur de la gestion des revenus personnels varie substantiellement d'un individu à l'autre, de même que la pertinence des achats effectués par les ménages. Cependant, nous laisserons ce vaste débat aux économistes et aux sociologues.

être quasiment nul. Ces exemples étaient destinés à faire prendre conscience aux sujets de la relativité et de la variabilité des prix et donc de rendre plus plausibles les prix fixés aléatoirement dans les jeux de données de test.

Nous pouvons à présent concrétiser nos questions abstraites sous la forme de tâches inscrites dans un scénario. La Table 27 en fournit quelques exemples basés sur la hiérarchie de la Fig. 156.

Table 27: Tâches scénarisées utilisées dans les tests sommatifs

Type	Tâche
HyVS	
	Identifier la suite des parents et ancêtres du nœud N <i>En remontant jusqu'à la catégorie générale 'Produits', dans quelles catégories successives de produits est classée la catégorie « Pommes » ?</i>
	Parmi les enfants du nœud N, identifier celui ayant le plus d'enfants <i>Parmi les sous-catégories de la catégorie de produits 'Hygiène', laquelle possède le plus de sous-catégories de niveau immédiatement inférieur ?</i>
	Compter le nombre de nœuds frères du nœud N <i>Dans la catégorie de produits « Viandes », combien existe-t-il de sous-catégories autres que les volailles ? (ne comptez pas la catégorie 'volailles' dans le total)</i>
HyVC	
	Classer n nœuds identifiés en ordre décroissant de pondération <i>Classez les catégories de produits suivantes : 'Fruits', 'Lessives' et 'Boissons' en ordre décroissant de budget.</i>
	Trouver le nœud ayant la pondération la plus faible parmi n nœuds identifiés appartenant au même niveau hiérarchique <i>Quel est le poste de budget le moins coûteux entre les catégories « Légumes », « Cosmétiques » et « Lessive » ?</i>
	Trouver le nœud ayant la pondération la plus forte parmi n nœuds identifiés n'appartenant pas au même niveau hiérarchique <i>Quel est le poste de budget le plus coûteux entre les catégories « Glaces », « Viandes » et « Cerises » ?</i>
HyVSC	
	Trouver le nœud ayant la pondération la plus forte parmi les parents de n nœuds identifiés <i>Parmi les catégories de produits de niveau immédiatement supérieur aux catégories suivantes : 'Desserts', 'Mocassins' et 'Lessives', laquelle nécessite le plus gros budget ?</i>
	Trouver le nœud ayant la pondération la plus faible parmi les enfants de n nœuds identifiés <i>Parmi les catégories de produits de niveau immédiatement inférieur aux catégories 'Boissons', 'Ménage' et 'Chaussures', laquelle nécessite le plus petit budget ?</i>
	Parmi les feuilles descendant du nœud N, trouver celle ayant la pondération la plus forte <i>Parmi les sous-catégories de produits descendant de la catégorie 'Habillement', identifiez celles qui n'ont pas de sous-catégories (ne vous limitez pas au niveau immédiatement inférieur à « Habillement »). Parmi les catégories que vous venez d'identifier, laquelle nécessite le plus gros budget ?</i>

c. *Mesures associées aux tâches*

A présent que les tâches à évaluer sont identifiées, il faut préciser les métriques qui serviront de base à leur évaluation. En matière d'évaluation, Shneiderman et Plaisant (2005, p. 162) préconisent d'employer des critères mesurables tels que :

- le temps nécessaire aux utilisateurs pour apprendre des fonctions spécifiques,
- la vitesse avec laquelle une tâche est réalisée,
- le taux d'erreur par utilisateur,
- la rétention des commandes par les utilisateurs au cours du temps,
- la satisfaction subjective des utilisateurs.

Ces critères concernent des objets de natures différentes. On distingue des critères associés à des tâches (vitesse, taux d'erreur), à des fonctionnalités du logiciel (temps d'apprentissage, rétention des commandes) et à une perception globale de l'application (satisfaction subjective). Nous commencerons par nous focaliser sur les mesures associées aux tâches. Nous discuterons ensuite d'une mesure globale. Nous avons choisi de ne pas inclure de critères directement liés à une fonction du logiciel car notre objectif principal est d'évaluer des techniques de visualisation. Comme il est difficile de les évaluer sans les implémenter en pratique dans une application informatique, les tests reposeront sur l'utilisation de notre prototype. Cependant, nous ne visons pas à ce niveau à évaluer la qualité de notre développement. De plus, malgré leur indéniable pertinence, les deux métriques liées aux fonctions du logiciel nécessitent une méthodologie basée sur des études longitudinales que nous ne pouvions mettre en œuvre en pratique.

Au niveau des tâches, nous avons donc opté pour deux variables : la vitesse pour réaliser une tâche et le taux d'erreur.

La vitesse pour réaliser une tâche sera calculée comme la différence de temps écoulé entre le moment où l'utilisateur commence la tâche et le moment où il déclare l'avoir terminée. En termes rigoureux, il ne s'agit donc pas d'une vitesse mais d'un temps. Nous adopterons donc le vocable : 'temps mis pour réaliser la tâche'.

Le taux d'erreur sera défini comme le nombre de tâches réalisées avec succès divisé par le nombre de tâches que le sujet du test devait réaliser pour une configuration particulière du test. Le fait de ne pas répondre à une question dans le temps imparti sera considéré comme un échec.

Nous avons en effet fixé une limite de temps pour réaliser les différentes tâches. Néanmoins, une valeur élevée a été choisie. Concrètement, elle a été déterminée comme un multiple (3 x) du temps maximum observé lors des évaluations de tests à blanc conduits préalablement aux tests réels. Cette approche se justifie par le fait que nous ne cherchons pas à évaluer une interface homme-machine destinée spécifiquement à être utilisée en situation de stress temporel (au contraire, par exemple, de l'instrumentation présente dans un cockpit d'avion de chasse). Il n'y a donc pas d'intérêt particulier à placer le sujet dans une telle situation.

4.1.5. Plan d'expérience

a. *Variables indépendantes*

En statistique, les variables indépendantes sont celles qui sont manipulées par l'expérimentateur alors que les variables dépendantes sont celles qui constituent les résultats de l'expérience. Comme le résume Howell (1998, p. 4), '*fondamentalement, l'étude porte sur les variables indépendantes et les résultats de l'étude (les données) sont les variables dépendantes*'.

D'un point de vue général, nous nous posons la question de savoir si les techniques de visualisation *treemap* et *ellimap* sont équivalentes pour aider l'utilisateur à percevoir la structure et le poids relatif des nœuds d'une hiérarchie. Sur base de cet objectif et des

différentes considérations détaillées dans les sections précédentes, nous identifions formellement deux variables indépendantes :

- la ‘technique de visualisation’ (V) qui peut prendre deux valeurs : ‘*squarified treemap*’ ou ‘ellimap avec algorithme de division dépendante’⁴⁴,
- le ‘type de tâche’ (T) évalué qui peut prendre trois valeurs : ‘Tâche Hybride Valeur – Structure’ (HyVS), ‘Tâche Hybride Valeur – Comparaison’ (HyVC) et ‘Tâche Hybride Valeur – Structure – Comparaison’ (HyVSC)

Nous avons choisi de nous limiter à ces deux variables.

La taille du jeu de données visualisé aurait également pu être considérée comme une variable indépendante car elle peut jouer un rôle important dans la performance offerte par une technique de visualisation.

Cependant, nos évaluations pilotes (cf. 4.2.1, p. 268) nous ont renseignés sur le temps nécessaire pour conduire correctement une séance d’évaluation basée sur les deux variables V et T. En moyenne, une séance complète durait environ une heure. Cette durée est apparue comme la borne supérieure acceptable pour les sujets issus de notre population cible. Des séances de test plus longues auraient généré divers problèmes organisationnels.

En introduisant, par exemple, une nouvelle variable ‘*Taille du jeu de données*’, nous aurions allongé la durée des tests au-delà de ce qui était acceptable. Deux options se présentaient donc à nous :

- ajouter cette nouvelle variable et trouver une nouvelle population dans laquelle tirer des sujets de tests ;
- ne pas intégrer cet élément comme une variable indépendante et conserver la population à laquelle nous avions prévu d’avoir accès.

Nous avons retenu la seconde solution car nous pensons que le choix des sujets de test est primordial. Comme il ne nous était pas possible de maîtriser ce paramètre avec rigueur dans un autre cadre que celui du centre de recherche, nous avons décidé de ne considérer qu’une seule taille pour le jeu de données de test.

Par contre, nous avons opté pour une taille représentative du contexte de notre travail de recherche : les plates-formes électroniques de coopération. Sur base de notre expérience et des cas réels d’utilisation qui ont sous-tendu notre réflexion, la taille du jeu de données a été fixée à environ 500 nœuds. La profondeur de l’arborescence a été fixée à 8 niveaux. Il s’agit d’une taille moyenne mais elle permet déjà de mettre en évidence certains problèmes liés au volume de données. Une telle structure arborescente ne peut plus être considérée comme triviale à visualiser.

Par ailleurs, nous pouvons aussi noter qu’un ensemble de données de quelques centaines de nœuds se rencontre également dans d’autres contextes applicatifs (p.ex. organigramme d’entreprise, structure d’un site web, ...). Les résultats de notre évaluation présentent donc une portée qui dépasse le cas strict des plates-formes coopératives.

b. Variables dépendantes

Il nous faut à présent définir les variables dépendantes.

Nous avons discuté dans la section 4.1.4 des observations que nous souhaitons collecter. Ces observations constituent la matière première des variables dépendantes. Trois variables dépendantes peuvent donc être définies.

Premièrement, le temps mis pour réaliser une tâche sera calculé en pratique par la moyenne des temps associés à la réalisation d’un ensemble de tâches jugées similaires. Nous nommerons cette variable ‘Temps moyen par tâche’ (Tm).

⁴⁴ Par souci de concision, nous nous limiterons à mentionner les termes ‘*treemap*’ et ‘*ellimap*’ dans la suite de la discussion sur les tests mais il faut garder à l’esprit quelles variantes de ces techniques de visualisation sont concernées.

La seconde variable dépendante sera le ‘Taux d’erreur’ (Err). Elle sera aussi calculée à partir d’un ensemble de tâches jugées similaires. Par exemple, si nous demandons au sujet de réaliser trois tâches associées à la perception de la structure des données, le Temps moyen par tâche T_m et le taux d’erreur T_{err} seront calculés pour chaque sujet sur base des résultats obtenus lors de la réalisation de ces trois tâches.

Les deux variables T_m et Err prennent des valeurs continues. T_m sera exprimé en secondes et Err en pourcentage.

La troisième variable dépendante sera une ‘évaluation subjective’ par l’utilisateur (Sbj). Elle prendra des valeurs discrètes (cf. échelle à n niveaux).

Dès lors que les variables indépendantes et dépendantes ont été identifiées et définies, nous devons nous interroger sur le plan d’expérience à mettre en œuvre pour collecter des données. En fait, nous sommes en présence de contextes semblables. Dans le premier cas, nous visons à déterminer si les variables indépendantes⁴⁵ V et T exercent une influence sur les variables dépendantes T_m et Err.

Dans le second cas, nous cherchons à déterminer si les variables indépendantes V et T jouent un rôle dans la valeur de la variable dépendante Sbj.

c. Plans factoriels

Pour les expériences associées aux variables dépendantes T_m et Err, nous sommes en présence de ce que l’on appelle en statistiques un plan factoriel complet. Howell (1998, p. 452) définit un tel plan comme *‘une expérience dans laquelle chaque niveau de chaque facteur est couplé à chaque niveau de chacun des autres facteurs. En d’autres termes, un plan factoriel est un plan qui inclut toutes les combinaisons des niveaux des variables indépendantes.’*

La Table 28 résume les plans factoriels que nous mettrons en œuvre lors des tests relatifs aux tâches.

Table 28: Résumés des plans factoriels pour les expériences liées aux tâches

Expérience	Variables indépendantes / Facteurs	Variables dépendantes
Tâches A	V, T (2 x 3)	T_m
Tâches B	V, T (2 x 3)	Err

Les plans factoriels des deux expériences relatives aux tâches sont identiques. Avec la notation classique, ils peuvent se résumer par l’expression : V x T. Ils incluent six combinaisons différentes de valeurs pour les deux variables indépendantes.

Une première question cruciale qui se pose à nous est la suivante : chaque sujet sera-t-il confronté à une seule technique de visualisation ou aux deux ? Selon la réponse que nous apporterons, les méthodes statistiques à appliquer seront différentes. Dans le premier cas, nous parlerons d’un test inter-sujets (*between-subjects*) et dans le second d’un test intra-sujets (*within-subject*).

Nous avons opté pour une configuration de test intra-sujets dans laquelle tous les participants au test utiliseront les deux techniques de visualisation pour réaliser chacun des trois types de tâches. Ce choix a été fait car cette configuration permet d’obtenir des résultats plus précis avec un nombre plus petit d’utilisateurs (cf. p. 240). De plus, elle évite les problèmes associés à la constitution de groupes équivalents inhérents à la configuration inter-sujets.

d. Echantillonnage

Dans une section précédente (cf. p. 254), nous avons discuté des caractéristiques de la population à prendre en compte dans notre évaluation comparative des ellimaps et des

⁴⁵ Dans ce contexte, les variables indépendantes sont appelées ‘facteurs’.

treemaps. Nous avons ensuite identifié les variables dépendantes et indépendantes que nous considérerons dans notre étude. Nous pouvons donc à présent nous consacrer à la problématique de l'échantillonnage.

Le but d'une technique d'échantillonnage est d'extraire un sous-ensemble fini d'objets (appelé échantillon) dans un ensemble plus large (appelé population) de telle sorte que l'échantillon soit représentatif de la population toute entière.

Avant toute autre considération, il faut rappeler que les méthodes statistiques utilisées dans ce cadre reposent sur l'hypothèse que le processus d'échantillonnage est un phénomène aléatoire. Le choix des sujets de test ne peut donc s'effectuer sur base de préférences personnelles ou de connaissances préalable de leurs opinions.

Trois grandes approches sont habituellement rencontrées pour répondre à ce défi :

- le tirage avec remise, dans lequel chaque individu a toujours la même probabilité d'être choisi ;
- le tirage sans remise, dans lequel on ne peut choisir qu'une seule fois chaque individu ;
- le tirage stratifié, dans lequel on fixe a priori à la fois des catégories pour classer les individus et le nombre d'individus à choisir dans chacune de ces catégories.

Afin de pouvoir appliquer les tests statistiques classiques, nous devons utiliser la méthode du tirage avec remise. En effet, ces tests se basent sur l'hypothèse de l'indépendance des observations. Cette contrainte détermine le mode de calcul de la variance et les formules utilisées dans la plupart des tests classiques se basent sur une expression de la variance pour observations indépendantes.

La procédure d'échantillonnage exige de disposer d'une population accessible dans laquelle il est possible de tirer au hasard des individus. Nous avons choisi de réaliser le tirage aléatoire des individus qui composeront l'échantillon au sein des membres du personnel d'un centre de recherche auquel nous avons accès. Ils répondent globalement aux caractéristiques de la population exigées pour notre évaluation.

Au moment où la procédure d'évaluation fut implémentée, le centre comptait environ 170 personnes. Elles se répartissent en quatre départements scientifiques et un département administratif. Les quatre départements de recherche sont actifs dans les disciplines suivantes :

- sciences de l'environnement (biologie, chimie, hydrologie) ;
- sciences des matériaux ;
- informatique ;
- technologies associées à l'industrie automobile (en particulier l'électronique).

Le département administratif regroupe des personnes travaillant dans différents services : comptabilité, gestion des ressources humaines, gestion des achats, services techniques, gestion de la bibliothèque... Ces personnes n'ont pas un profil scientifique.

De manière générale, les membres du personnel du centre de recherche disposent tous d'une connaissance minimale des outils informatiques. En ce qui concerne la connaissance des techniques de visualisation d'information, il n'est pas impossible que quelques rares personnes possèdent une certaine expertise mais il est clair que cette discipline est inconnue de la vaste majorité des employés.

Le niveau de formation correspond à nos contraintes puisque la totalité des membres du personnel dispose d'au moins un diplôme de l'enseignement secondaire supérieur (Bac). Les disciplines dans lesquelles les diplômes ont été obtenus sont très variables (biologie, chimie, informatique, ingénierie, comptabilité, secrétariat, philologie) mais il faut reconnaître que les formations scientifiques sont sur-représentées par rapport à une population classique.

Les professions regroupées au sein du centre sont, elles aussi, multiples. Outre la présence de chercheurs, nous rencontrons également des comptables, des secrétaires ou encore des personnes chargées du service technique.

Les différentes catégories d'âge d'une population adulte classique sont présentes dans le centre. Les plus jeunes ont une vingtaine d'années (stagiaires) et les plus âgés sont quinquagénaires.

Enfin, le centre compte à la fois des hommes et des femmes en proportions quasiment égales.

Sur base de ces différents paramètres, la population du centre de recherche considéré nous est apparue acceptable pour en extraire un échantillon.

La question qui se pose dès lors est celle du nombre de sujets participant à l'évaluation. L'étude de la littérature nous indique qu'un échantillon d'une vingtaine de personnes est un minimum. Pour des études comparables à la nôtre, nous avons noté que :

- O'Donnel et al. (2006) utilisent 16 sujets,
- Demian et Fruchter (2006b) utilisent 20 sujets,
- Andrews (2006, 2007) utilise 32 sujets.

Pour notre part, nous nous sommes basés sur les théories statistiques pour estimer le nombre de sujets nécessaires pour faire apparaître une différence significative entre les techniques de visualisation que nous évaluons. Il se base sur les observations collectées pendant les expériences pilotes qui ont précédé les tests réels (cf. 4.2.1, p. 268).

Puisque nous cherchons à tester des hypothèses relatives à l'égalité de moyennes entre échantillons non indépendants, Dagnelie (2006, p. 234-235) explique que la détermination du nombre d'observations peut être réalisée à l'aide de la relation : $n = (u_{1-\alpha/2} + u_{1-\beta})^2 \sigma^2 / \delta^2$ pour des effectifs au moins égaux à 10 et des risques⁴⁶ de première et de deuxième espèce α et β inférieurs ou égaux respectivement à 0,1 et 0,5 ; σ^2 désignant la variance des différences et δ la différence de moyennes que l'on souhaite mettre en évidence.

Sur base de nos expériences pilotes (cf. p. 268), nous avons estimé les valeurs des paramètres comme suit : $\delta = 15$ sec et $\sigma = 25$ sec pour les temps d'accomplissement des tâches (indépendamment du type de tâche).

Le risque de première espèce jugé acceptable a été fixé à 5% ($\alpha = 0,05 \Rightarrow u_{0,975} = 1,96$) et le risque de seconde espèce à 10% ($\beta = 0,10 \Rightarrow u_{0,90} = 1,28$).

Sur base des valeurs attribuées à ces paramètres, 30 personnes sont nécessaires pour réaliser notre test. Cependant, afin de nous prémunir d'éventuels problèmes survenant au cours des tests, nous avons pris une marge de sécurité en décidant de fixer le nombre de participants à 35 sujets.

Cette précaution fut judicieuse. En effet, un des tests fut interrompu de manière inattendue parce que le sujet fut appelé en urgence dans son service. Dès lors, les résultats collectés au cours de cette séance furent simplement annulés. Il reste donc 34 sujets dans les résultats finaux.

La composition de l'échantillon est résumée dans les tableaux ci-dessous.

Table 29: Composition de l'échantillon : Genre

Genre	Nombre absolu d'individus	Nombre relatif d'individus
Hommes	22	64,7 %
Femmes	12	35,3 %

⁴⁶ Pour rappel, en inférence statistique, le risque de première espèce α est celui de rejeter l'hypothèse nulle alors qu'elle est vraie et le risque de seconde espèce β est celui d'accepter l'hypothèse nulle alors qu'elle est fausse (cf. [45], p. 412). Le niveau de signification d'un test correspond au risque de première espèce.

Table 30: Composition de l'échantillon : Age

Age	Nombre absolu d'individus	Nombre relatif d'individus
0-19 ans	1	2,9 %
20-29 ans	17	50 %
30-39 ans	15	44,1 %
40-49 ans	1	2,9 %
50-59 ans	0	0 %

Table 31: Composition de l'échantillon : Niveau de diplôme

Niveau de diplôme	Nombre absolu d'individus	Nombre relatif d'individus
Bac	2	5,9 %
Bac + 2/3	6	17,6 %
Bac + 4/5	20	58,8 %
Doctorat	6	17,6 %

Table 32: Composition de l'échantillon : Diplôme

Diplôme	Nombre absolu d'individus	Nombre relatif d'individus
Sciences naturelles	13	38,2 %
Sciences humaines	0	0 %
Sciences médicales	0	0 %
Sciences de l'ingénieur	7	20,6 %
Informatique et télécoms	10	29,4 %
Sciences économiques	2	5,9 %
Droit	0	0 %
Autres	2	5,9 %

Table 33: Composition de l'échantillon : Profession

Profession	Nombre absolu d'individus	Nombre relatif d'individus
Chercheur en informatique	7	20,6 %
Technicien informatique	1	2,9 %
Chercheur (autres disciplines que l'informatique)	14	41,2 %
Technicien de laboratoire	5	14,7 %
Comptable	1	2,9 %
Attaché aux ressources humaines	2	5,9 %
Assistant(e) de communication	2	5,9 %
Attaché au service achats	1	2,9 %
Gestion administrative	1	2,9 %

Tous les sujets possédaient une expérience de plus d'un an avec l'informatique.

En ce qui concerne la connaissance de techniques ou de logiciels de visualisation, les résultats obtenus dans le questionnaire de profil ne nous ont pas semblé suffisamment réalistes pour être pris en compte. Plusieurs séances de débriefing ont en effet révélé des incohérences entre les réponses du questionnaire et la discussion. D'un point de vue global, nous pouvons cependant affirmer que la majorité des sujets connaissaient les techniques classiques du type '*pie chart*' ou 'histogrammes'. De même, des logiciels standards tels que Microsoft Excel étaient connus. Par contre, très peu de sujets disposaient d'une connaissance, même sommaire en visualisation d'information, au niveau théorique ou au niveau de logiciels dédiés.

4.2 Mise en œuvre du plan d'expérience

4.2.1. Expériences pilotes

Comme cela est fortement recommandé, nous avons commencé par réaliser des tests pilotes qui ont servi à valider la procédure de test. D'un point de vue méthodologique, cette évaluation à blanc est essentielle car elle permet d'identifier d'éventuelles lacunes ou imprécisions dans l'élaboration de la procédure expérimentale. Celles-ci peuvent ensuite être corrigées pour éviter de compromettre la validité de l'entièreté des tests.

Par ailleurs, les tests pilotes ont également permis de calculer des estimations utiles pour les tests statistiques (cf. nombre de sujets nécessaires).

Les évaluations pilotes ont été réalisées avec trois utilisateurs : un informaticien, une bibliothécaire et un responsable de l'infrastructure technique (bâtiment). Aucun des trois n'est expert en visualisation d'information.

Ce test a permis de mettre en évidence certains éléments du test qui devaient être améliorés.

- Vu le stress observé chez les utilisateurs au cours des tests à blanc, nous avons constaté qu'il est essentiel d'insister sur le fait que c'est le logiciel qui est évalué et non la personne qui participe au test.
- Pour éviter d'induire un stress chez les sujets, nous avons noté qu'il faut bien les informer du fait qu'ils disposent du temps nécessaire pour répondre aux questions et que le but de l'exercice n'est pas d'évaluer leur performance sous une contrainte de temps exigeante.
- Certaines questions étaient difficiles à comprendre, voire confuses. Elles ont été reformulées.
- Dans notre présentation générale des concepts et de l'application évaluée, nous avons ajouté une explication sur le fait que les couleurs de remplissage des formes imbriquées (ellipses / rectangles) ne communiquaient pas de sémantique particulière. Elles sont présentes pour faciliter la perception des différents niveaux hiérarchiques.
- Nous avons constaté que les sujets ne connaissent pas la discipline de la visualisation d'information et qu'il est difficile, voire illusoire d'évaluer leur connaissance précise à ce niveau. Nous avons donc inclus dans le questionnaire de profil des questions concernant ce point plus simples que celles prévues initialement.

Ces expériences pilotes ont aussi fourni des données de base pour calibrer la procédure de test réelle, telles que le temps limite accordé pour répondre à une question. Dans notre cas, le temps maximum observé pour réaliser une tâche dans l'expérience pilote était d'environ 90 secondes. Comme nous ne souhaitons pas placer les sujets en condition de stress temporel, nous l'avons triplé pour fixer le temps limite maximum autorisé lors des tests réels. Nous avons donc obtenu une borne supérieure de 5 minutes pour réaliser les tâches dans l'évaluation réelle. Cette valeur correspond en pratique à l'absence de contrainte temporelle.

4.2.2. Déroulement des tests réels

Nous discutons dans cette section de la procédure de test, c.-à-d. de la description du déroulement d'une session de test. Nous avons adopté l'approche classique qui consiste à diviser le test en quatre parties :

- une introduction,
- la réalisation de tâches (via des scénarios),
- des activités postérieures à l'accomplissement des tâches,
- un débriefing.

On peut rappeler qu'une séance de test concerne un seul sujet à la fois.

a. *Partie 1 : Introduction*

La phase d'introduction regroupe différentes opérations.

Pour commencer, les évaluateurs doivent expliquer le but du test. A ce niveau, il est essentiel de faire comprendre aux sujets que l'évaluation porte sur la technique de visualisation et pas sur leur propre personne. Nous avons donc insisté sur ce point.

Table 34: Questionnaire de détermination du profil des sujets

Genre :	<i>homme / femme</i>
Quel est votre âge ?	<i>moins de 20 ans / entre 20 et 29 ans / entre 30 et 39 ans / entre 40 et 49 ans / entre 50 et 59 ans / 60 ans et plus</i>
Quel est le plus haut diplôme dont vous disposez ?	<i>Bac ou équivalent / Bac+2/3 / Bac+4/5 / Doctorat</i>
Dans quelle(s) discipline(s) êtes-vous diplômé(e) ?	<i>Sciences naturelles (biologie, physique, chimie...) / Sciences humaines (langues, littérature, communication...) / Sciences médicales / Sciences de l'ingénieur (électricité, mécanique, construction...) / Informatique et télécommunications / Sciences économiques (finances, comptabilité, marketing...) / Droit / Autre : précisez.</i>
Quelle profession exercez-vous ?	<i>précisez</i>
Depuis combien de temps utilisez-vous l'informatique (quel que soit le contexte : privé ou professionnel) ?	<i>moins de trois mois / entre trois mois et un an / plus d'un an</i>
Connaissez-vous des outils informatiques pour représenter graphiquement de l'information ?	<i>Oui / Non. Si oui, citez des exemples :</i>
Avez-vous déjà utilisé des outils informatiques pour représenter graphiquement de l'information ?	<i>Oui / Non. Si oui, lesquels ?</i>
J'accepte que les données relatives à ce test soient utilisées à des fins scientifiques. Le caractère anonyme de celles-ci est assuré.	

Ensuite, nous notifions au sujet les éléments éthiques habituels tels que la possibilité d'arrêter le test à tout moment à leur demande ou le respect de la confidentialité des résultats nominatifs.

Puis, il est demandé au sujet de remplir un questionnaire permettant de déterminer son profil (cf. Table 34). Ce questionnaire fait référence aux éléments discutés dans la section 4.1.3 (p. 254).

Le niveau d'expérience avec les *treemaps* et les *ellimaps* n'est pas inclus dans le questionnaire de profil car les questions y relatives sont insérées dans le questionnaire d'évaluation subjective (questionnaire de satisfaction) proposé après la réalisation des tâches.

Le questionnaire regroupant les différentes variables est présenté ci-dessous.

Il faut noter à la fin du questionnaire la présence d'une mention qui précise explicitement l'acceptation par le sujet de l'utilisation des données à des fins scientifiques.

L'étape suivante consiste à dispenser au sujet une formation rapide aux concepts de visualisation par *treemap* et *ellimap*. Cette formation débute par une démonstration du logiciel utilisé pour les tests. Au cours de cette étape, la manipulation du logiciel est prise en charge par un évaluateur. Ensuite, le sujet est invité à répondre à cinq questions similaires à celles du test réel en utilisant l'application de test. L'évaluateur présent à ses côtés répond aux questions du sujet et l'aide à bien s'approprier l'environnement de test.

L'étape d'apprentissage porte donc à la fois sur les techniques de visualisation évaluées et sur la manière de répondre aux questions lors du test.

Pendant toute cette phase d'introduction, les sujets sont libres de poser toutes les questions qu'ils souhaitent pour bien comprendre ce que nous attendons d'eux.

Enfin, juste avant de commencer la seconde partie du test, l'évaluateur rappelle explicitement les conseils suivants pour faire prendre conscience des erreurs habituelles rencontrées au cours des expériences pilotes :

- « *Quand une nouvelle question apparaît, la visualisation de la hiérarchie est toujours réinitialisée. La catégorie la plus haute est toujours celle des 'produits'.* »
- « *Toutes les catégories ne sont pas visualisées. Le niveau de profondeur de l'affichage est limité. Vous devrez donc parfois naviguer dans la hiérarchie pour pouvoir répondre correctement à une question.* »
- « *Les noms des catégories qui sont mentionnés dans les questions sont exactement ceux qui sont utilisés dans la visualisation. N'interprétez pas ou ne modifiez pas ces termes pour répondre aux questions.* »
- « *Même si des questions vous semblent similaires, les réponses peuvent être différentes car les hiérarchies visualisées peuvent différer.* »
- « *Veillez à bien lire et à bien comprendre la question avant d'essayer de répondre !* »
- « *Vous pouvez passer à la question suivante sans répondre.* »

b. Partie 2 : Réalisation de tâches

La seconde partie du test est celle qui exige le plus de temps. Elle vise à faire réaliser différentes tâches aux sujets.

Au cours de cette partie, les interactions entre le sujet et l'évaluateur sont limitées au strict minimum. En aucun cas, nous n'aidons le sujet à répondre à réaliser une tâche. Le sujet est seul dans la pièce des tests. Il est cependant filmé par une *web cam* en toute connaissance de cause. Ce dispositif étant uniquement destiné à informer l'évaluateur en cas de survenance d'un problème majeur.

Nous avons discuté précédemment du nombre de variables indépendantes et nous avons expliqué que six configurations différentes sont considérées. Comme chaque valeur de la variable T correspond en pratique à trois tests (cf. Table 26), chaque sujet devra réaliser dix-huit tâches au total.

Afin d'éviter un éventuel effet d'apprentissage ou de fatigue, il est nécessaire d'éviter un ordre constant des conditions testées pour tous les sujets. Dans notre cas, nous devons donc

veiller à ne pas proposer le même ordre pour nos deux conditions : (technique de visualisation V, type de tâche T).

Différentes méthodes permettent d'éviter ce type de biais.

Une première méthode est appelée 'carrés latins'. Elle consiste à opérer des permutations dans l'ordre des conditions pour éviter que certaines d'entre elles ne soient sur-représentées à une certaine place dans les évaluations. Elle fut par exemple utilisée par Ball et al. (2007) pour évaluer des méthodes de navigation associées à des dispositifs d'affichage de grande taille et par Robertson et al. (2005) pour évaluer une nouvelle technique de visualisation des correspondances entre schémas XML.

Nous avons choisi une autre méthode qui consiste à ordonner aléatoirement les conditions lors de leur administration aux sujets. Cette approche fut notamment utilisée par les auteurs d'évaluations comparables à la nôtre (p.ex. O'Donnell et al., 2006 ; Demian et Fruchter, 2006b). Pour mettre en œuvre cette contrainte, nous utiliserons un générateur de nombre pseudo-aléatoires qui est une des méthodes de randomisation suggérée par Dagnelie (1998, p. 111) dans son traité de statistiques.

c. Partie 3 : Questionnaire de satisfaction

Nous avons souhaité inclure dans notre plan d'expérience une mesure plus globale qui rend compte de la manière dont les deux techniques sont perçues par les utilisateurs. Nous adoptons ainsi une approche similaire à celle d'Andrews (2007) qui a également complété son évaluation formelle par des mesures de l'appréciation subjective des techniques de visualisation par les sujets du test.

La troisième partie du test vise donc à faire remplir un questionnaire de satisfaction destiné à collecter l'impression subjective des sujets par rapport aux techniques évaluées.

A cette fin, nous utiliserons un questionnaire composé de deux parties. La première a pour objectif de déterminer si le sujet connaissait les techniques évaluées préalablement au test. La seconde vise à évaluer l'impression subjective des sujets concernant la capacité des deux techniques de visualisation à permettre la réalisation de certaines tâches.

Comme pour les évaluations basées sur des tâches, nous considérons les variables indépendantes V et T pour l'évaluation subjective. Par contre, pour faciliter la compréhension du questionnaire de satisfaction par les sujets, nous ne considérons que deux valeurs pour la variable T : '*Tâches de classification*' et '*Tâches de perception du poids des nœuds*'. Pour la variable 'V', les deux valeurs '*Treemap*' et '*Ellimap*' sont conservées.

Le questionnaire utilisé lors de cette évaluation est relativement court. Comme il est présenté après les exercices portant sur l'évaluation des tâches, il nous a semblé important de limiter la durée nécessaire pour le remplir afin de réduire au maximum l'effet de lassitude ou de fatigue qui pourrait se manifester chez les sujets.

Deux questions sont destinées à estimer la connaissance préalable du sujet vis-à-vis des techniques de visualisations évaluées :

- *Avant de participer à ce test, connaissiez-vous la représentation basée sur des rectangles ?*
- *Avant de participer à ce test, connaissiez-vous la représentation basée sur des ellipses ?*

Pour répondre à ces questions, nous utiliserons une échelle à trois niveaux :

<i>Je ne connaissais pas cette technique</i>
<i>Je connaissais cette technique mais je ne l'avais jamais utilisée</i>
<i>J'avais déjà utilisé cette technique</i>

En ce qui concerne la satisfaction subjective des utilisateurs, nous utiliserons une échelle à cinq niveaux.

<i>Très mauvais</i>	<i>Mauvais</i>	<i>Moyen</i>	<i>Bon</i>	<i>Très bon</i>

Nous demanderons au sujet de coter les différentes techniques sur cette échelle.

Les questions utilisées sont les suivantes :

- *De manière générale, comment jugez-vous la représentation basée sur des rectangles ?*
- *De manière générale, comment jugez-vous la représentation basée sur des ellipses ?*
- *Lorsqu'il s'agit de répondre à une question financière (budgets) sur les catégories de produits, comment jugez-vous la représentation basée sur des rectangles ?*
- *Lorsqu'il s'agit de répondre à une question financière (budgets) sur les catégories de produits, comment jugez-vous la représentation basée sur des ellipses ?*
- *Lorsqu'il s'agit de répondre à une question sur la classification des catégories de produits, comment jugez-vous la représentation basée sur des rectangles ?*
- *Lorsqu'il s'agit de répondre à une question sur la classification des catégories de produits, comment jugez-vous la représentation basée sur des ellipses ?*

D'un point de vue méthodologique, nous pensons avoir limité fortement le biais lié à l'empathie des sujets pour les évaluateurs pour deux raisons. D'une part, la grande majorité des sujets ne connaissaient aucune des deux techniques et ne pouvaient dès lors pas savoir laquelle était proposée par les évaluateurs. D'autre part, les évaluateurs ne pouvaient se prévaloir d'aucune relation de supériorité hiérarchique avec les sujets.

d. Partie 4 : Débriefing

La séance de test se termine par un débriefing au cours duquel nous remercions le sujet d'avoir participé à l'évaluation. Nous lui demandons également de ne pas divulguer de détails sur la procédure ou les questions posées afin d'éviter d'induire des biais dans les séances d'évaluation avec d'autres personnes qu'il pourrait connaître.

Le débriefing inclut également une discussion ouverte sur des questions que le sujet souhaite poser sur le test ou sur des remarques éventuelles qu'il pourrait formuler. Ceci nous offre une occasion de collecter de l'information informelle sur les techniques évaluées et sur la procédure d'évaluation.

4.2.3. Infrastructure technique de base

D'un point de vue logistique, notre évaluation formelle repose sur l'utilisation de différents équipements.

Tout d'abord, il faut noter que nous disposons d'un environnement spécifique dédié aux tests. Il est situé dans un local isolé des bureaux. Nous ne risquons donc pas de voir les tests perturbés par des appels téléphoniques ou des visites imprévues.

Dans ce local, nous utilisons un poste de travail spécifique composé :

- d'un ordinateur de bureau (*desktop*) disposant d'un processeur Intel Core 2 à 2,40 GHz, de 2 GB de mémoire RAM et opérant sous Windows XP 5.1 SP2 ; il est équipé avec un clavier, une souris et un écran plat de type Eizo FlexScan S1701 en résolution 1280 x 1024 ;
- d'une webcam de type Logitech Sphere MP sur pied, offrant un angle de prise de vue plus large que les webcams habituelles.

Sur ce poste sont installés les deux logiciels utilisés au cours de nos séances de tests.

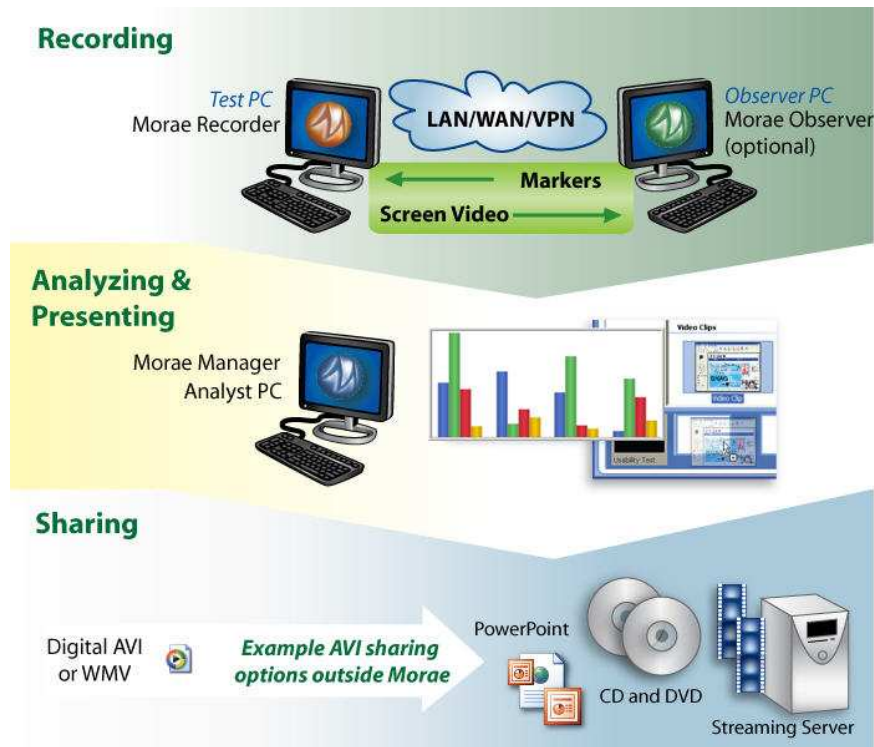


Fig. 157: Principe de fonctionnement de la suite Morae (source: TechSmith corp.)

Premièrement, nous utilisons la suite Morae 2.0 de la société TechSmith (cf. Fig. 157). Il s'agit d'un des utilitaires de référence pour la conduite de tests d'évaluation de logiciels. Elle se compose de divers modules :

- *Morae Recorder* qui permet d'enregistrer toutes les interactions entre le sujet et l'ordinateur, ainsi que les événements se produisant sur celui-ci ;
- *Morae Observer* qui permet d'observer (éventuellement à distance) le déroulement du test et si nécessaire de l'annoter ;
- *Morae Manager* qui est employé pour le paramétrage, la gestion et le post-traitement des enregistrements ;
- *Morae Player* qui permet de rejouer l'enregistrement d'une séance de test.

La suite Morae permet également d'inclure un questionnaire à la fin d'une séance de test. Nous avons utilisé cette fonctionnalité pour dispenser le questionnaire de satisfaction.

La suite Morae offre de nombreux avantages pour automatiser la collecte d'informations sur un test. Par contre, elle n'inclut évidemment pas le logiciel à évaluer.

En règle générale, on utilise pour les tests un logiciel donné sans y apporter de modifications. Dans notre cas, la situation se présente différemment. En effet, notre objectif consiste à évaluer de manière comparative deux techniques de visualisation. Nous nous focalisons plus sur les concepts à la base des techniques que sur une implémentation particulière (même si cette dernière est indispensable en tant que matérialisation d'un certain concept).

4.2.4. Environnement spécifique de test : FESTIV

Malgré ses avantages, l'infrastructure de test Morae ne suffisait pas pour mettre en œuvre les tests d'évaluation comparative comme nous le souhaitions. Dans ce contexte, nous étions particulièrement attentifs aux remarques formulées par Andrews (cf. section 4.1.1, p.252). En fait, si on souhaite comparer deux interfaces utilisateurs sur certains points spécifiques, il est important de neutraliser le plus possible les autres variables indépendantes.

Dès lors, nous avons décidé de concevoir un environnement spécifique dédié à nos tests. Nous l'avons appelé 'FESTIV' (*Formal Evaluation System for Testing Information Visualization*).

Si quelques exemples d'approches comparables existent pour conduire des expériences contrôlées dans le domaine de l'interaction homme-machine (cf. p. ex. Mackay et al., 2007), nous n'avons recensé dans la littérature qu'une seule application spécifique à la visualisation d'information. Il s'agit du système '*Hierarchical Visualisation Testing Environment*' (HVTE) décrit par Andrews (2007).

Parmi les avantages de l'utilisation d'une application unique, figure en bonne place le fait que les techniques de visualisation évaluées sont présentées dans un contexte tout à fait comparable. Hormis la technique elle-même, tous les autres paramètres d'interaction avec le système sont identiques. Par exemple, dans FESTIV, la navigation dans la hiérarchie, l'affichage de '*tooltips*' ou encore la politique de coloriage sont absolument identiques pour l'implémentation des deux techniques à comparer, à savoir les *treemaps* et les *ellimaps*. De cette manière, les biais liés à la comparaison d'environnements différents sont évités.

De plus, l'utilisation de FESTIV (ou de HVTE) permet de mieux contrôler le déroulement opérationnel du test et reproduire plus rigoureusement les différentes séances de test.

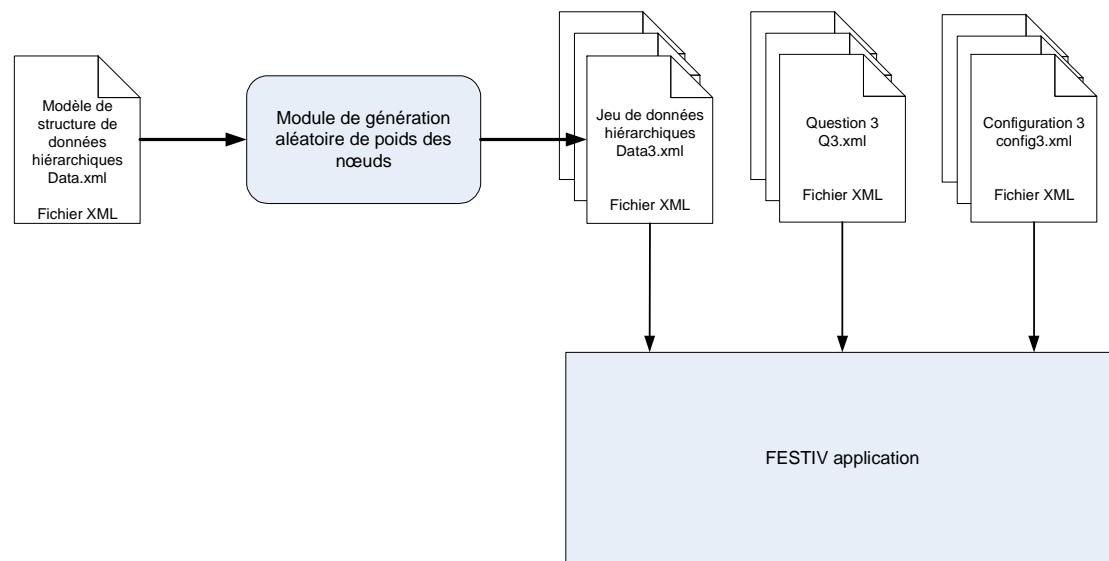


Fig. 158: Principe de fonctionnement du logiciel FESTIV

Bien qu'ils aient été développés indépendamment, HVTE et FESTIV reposent sur des principes similaires : les deux systèmes sont destinés à réaliser des évaluations formelles comparatives de techniques de visualisation d'information et leur mise en œuvre est comparable.

Une application unique est démarrée au début du test. Au sein de celle-ci, le sujet du test se voit proposer de réaliser une séquence de tâches portant sur différents jeux de données, eux-mêmes représentés avec différentes techniques de visualisation. L'application garantit que la séquence de tâches est différente pour chaque sujet en utilisant une technique adéquate (carrés

latins pour HVTE, générateur de nombres pseudo-aléatoires pour FESTIV). L'application enregistre les réponses aux questions posées ainsi que le temps mis pour répondre.

A ce stade, il est nécessaire d'attirer l'attention sur les différences existant entre le logiciel de test FESTIV et le prototype de visualisation de hiérarchies que nous avons conçu (cf. Chapitre 9 : Prototype 'Ellimaps', p. 291).

FESTIV utilise certains modules du prototype de visualisation complet, comme par exemple le positionnement spatial des ellipses dans les ellimaps ou des rectangles dans les *treemaps*. Par contre, compte tenu de sa vocation à éliminer tout biais dans l'évaluation, un nombre significatif de fonctions du prototype ne sont pas intégrées dans FESTIV. Par exemple, dans notre logiciel de visualisation de hiérarchies, une option existe pour coupler les vues *treemap* et ellimap avec une vue en arbre (cf. p. 292). Si cela s'avère être une fonctionnalité très intéressante en pratique, elle devient une source importante de biais si l'objectif est de comparer la lisibilité ou la navigation dans les deux premières vues. En effet, il serait très difficile de déterminer rigoureusement si l'utilisateur utilise une des deux vues en priorité ou plutôt la fonctionnalité de couplage dynamique, voire les deux vues indifféremment pour effectuer une tâche donnée.

Par ailleurs, FESTIV possède un ensemble de fonctionnalités qui ne présentent pas d'intérêt dans le prototype de visualisation de hiérarchies, telles que la présentation de questions en ordre aléatoire ou l'enregistrement des réponses de l'utilisateur de test.

La Fig. 158 illustre le principe de fonctionnement du logiciel FESTIV.

Un modèle générique de structure de données hiérarchique est utilisé en entrée du logiciel. Ce modèle est matérialisé par un fichier en format XML (*Data.xml*). Ensuite, un module de génération aléatoire du poids des nœuds produit à partir du fichier générique une série d'instances de jeu de données (*Data1.xml*, *Data2.xml*, *Data3.xml*). Les questions posées au sujet sont elles aussi introduites dans le logiciel FESTIV sous la forme de fichiers XML (*Q1.xml*, *Q2.xml*, *Q3.xml*). Elles sont par contre définies de manière statique par l'évaluateur. Enfin, des fichiers de configuration (*config1.xml*, *config2.xml*, *config3.xml*) sont utilisés pour préciser quelles fonctionnalités et quels composants graphiques sont présents dans l'interface utilisateur.

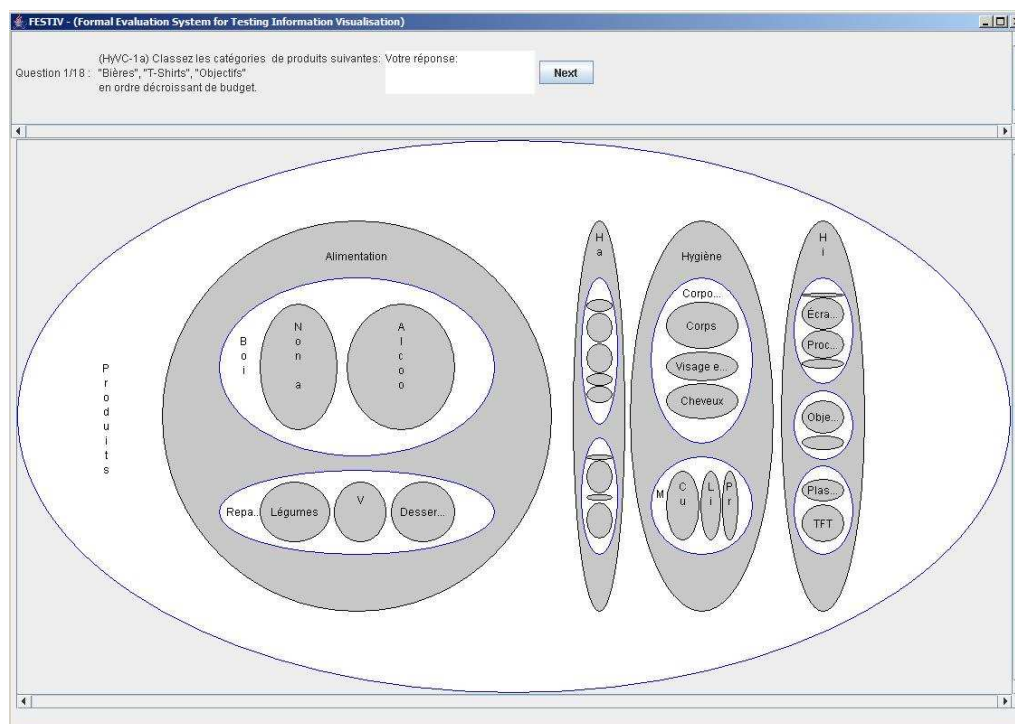


Fig. 159: Exemple de tâche à réaliser par le sujet dans l'environnement FESTIV (ellimap)

L'interface utilisateur du logiciel FESTIV est présentée dans les figures suivantes. Elle inclut un nombre très limité de fonctionnalités afin d'éviter autant que possible les biais dans la collecte des données expérimentales. Cette restriction est définie dans les fichiers de configuration. Il faut noter que, si nécessaire, une modification triviale de ceux-ci permettrait aisément d'inclure un des composants graphiques présents dans l'application complète (cf. p. 291) tel que le curseur qui ajuste dynamiquement la profondeur d'affichage ou la vue en arbre couplée au *treemap* / ellimap.

La plus grande partie de la fenêtre applicative est occupée par la vue à évaluer (cf. Fig. 159 pour ellimap ou Fig. 160 pour *treemap*).

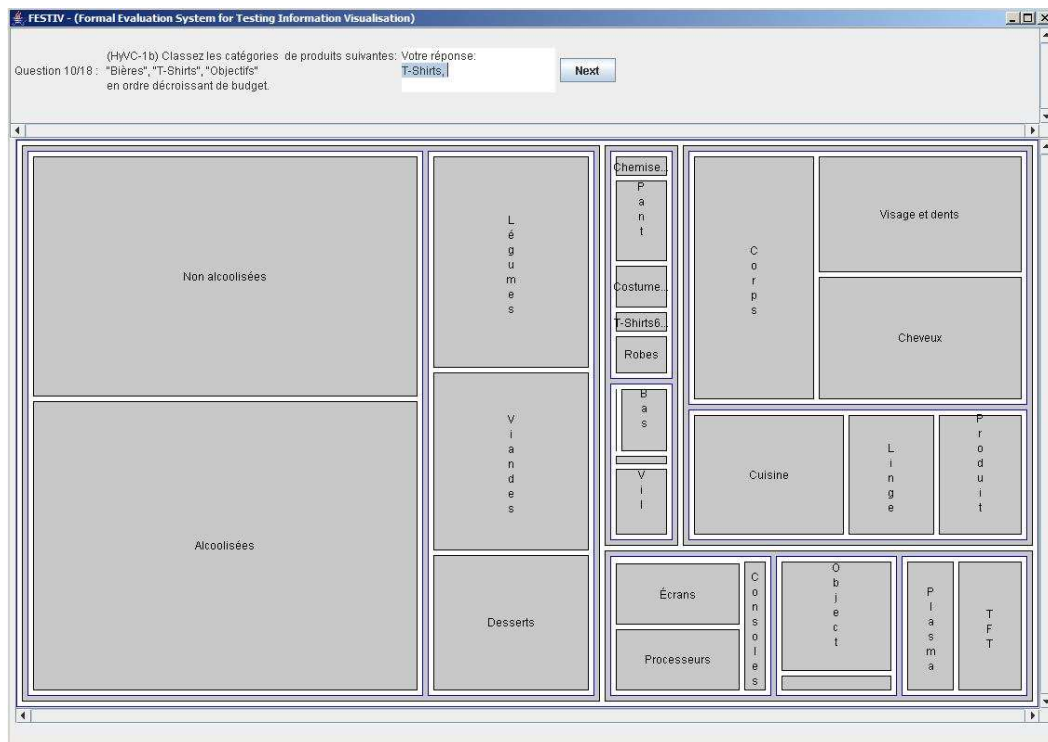


Fig. 160: Exemple de tâche à réaliser par le sujet dans l'environnement FESTIV (*treemap*)

La valeur du poids des nœuds est représentée par la surface de la forme géométrique associée mais elle est également accessible en déplaçant le curseur de la souris sur celle-ci. Cette opération fait apparaître un *tooltip* indiquant la valeur numérique du poids (cf. Fig. 161).

La navigation dans les deux vues *treemap* et ellimap est implémentée de manière absolument identique. En cliquant sur une ellipse ou un rectangle quelconque, l'utilisateur modifie la vue de telle sorte que cette forme géométrique devienne la racine (visuelle) de la hiérarchie affichée. La navigation ascendante est implémentée sous la forme d'un clic droit de la souris faisant apparaître une possibilité de choix entre remonter d'un seul niveau ou jusqu'à la racine du jeu de données (cf. Fig. 162).

La profondeur d'affichage a été fixée à 4 niveaux à partir du nœud courant. Elle ne peut être modifiée par le sujet.

Dans la zone supérieure de la fenêtre est présentée une question qui demande de réaliser une tâche donnée. La position de la question dans la liste complète est mentionnée afin de permettre au sujet de situer son état d'avancement dans le test. Une zone permettant l'inscription de la réponse est placée à côté de la question. Enfin, un unique bouton permet de passer à la question suivante. Il faut noter qu'il est interdit de revenir en arrière dans la liste des questions. Un système de temporisation déclenche automatiquement le passage à la question suivante après une durée paramétrable.

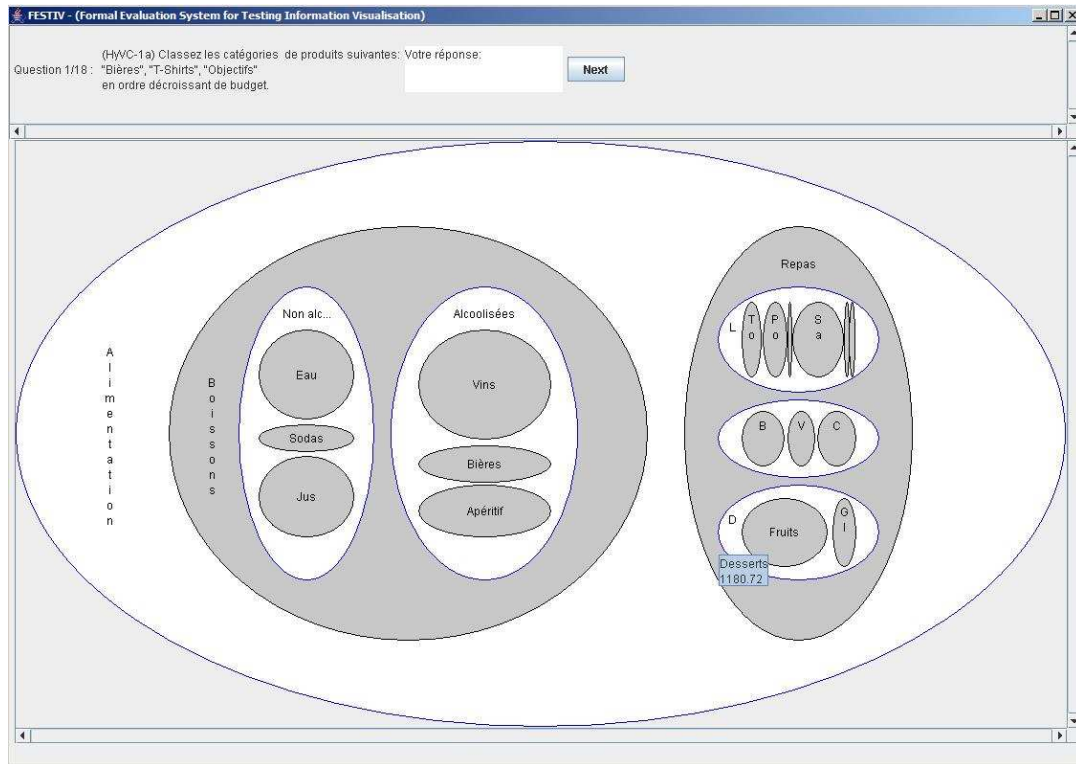


Fig. 161: Affichage dynamique d'étiquette

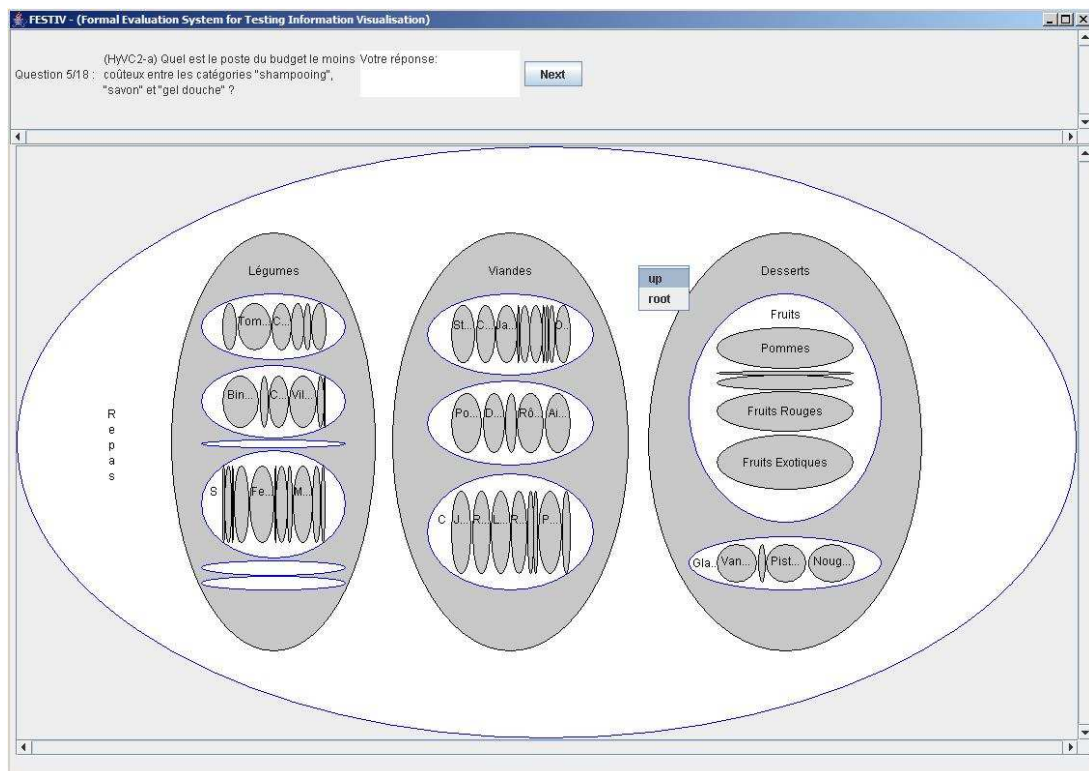


Fig. 162: Fonctionnalité de navigation ascendante

4.3 Résultats

4.3.1. Taux d'erreur et temps de réponse

a. Résultats bruts

Pour rappel, nos tests concernent deux variables dépendantes : le taux d'erreur (Err) et le temps moyen par tâche (Tm) et deux variables indépendantes : la technique de visualisation (V) et le type de tâche (T).

En pratique, la variable 'taux d'erreur' (Err) est définie comme le ratio entre le nombre de tâches correctement réalisées (n_{correct}) et le nombre de tâches à réaliser. Pour chaque type de tâche (HyVS, HyVC, HyVSC), trois questions correspondant à trois tâches individuelles ont été posées. Par conséquent, le taux d'erreur est défini par l'expression : $\text{Err} = n_{\text{correct}} / 3$. A ce stade, il faut mentionner qu'aucun sujet n'a dépassé le temps imparti pour répondre à une question. Les erreurs ne concerneront donc que les cas où le sujet a fourni une réponse erronée. Notre souhait de ne pas imposer de limite au temps de réponse a donc été satisfait en pratique.

En ce qui concerne la variable Tm, la valeur prise en considération dans l'analyse statistique est la moyenne des temps mis par le sujet pour réaliser l'ensemble des tâches du type considéré (HyVS, HyVC ou HyVSC). De plus, les temps de réponse associés à des réponses erronées sont exclus du calcul des moyennes. Ils sont codés comme des valeurs inexistantes.

Les résultats bruts sont présentés dans la Table 35.

La question fondamentale qui se pose à nous consiste à tester nos hypothèses H_i , compte tenu des résultats obtenus dans différentes conditions expérimentales. En d'autres termes, nous devons appliquer des méthodes statistiques pour déterminer si l'ellimap est meilleure que le *treemap* pour réaliser certaines tâches particulières. Plus précisément, nous chercherons à tester si les moyennes des taux d'erreur Err et des temps moyens par tâche Tm pour notre échantillon de 34 personnes sont significativement différents dans les différentes conditions.

Table 35: Résultats bruts de l'évaluation comparative

		Treemap		Ellimap	
		Moyenne μ	Dév. std. σ	Moyenne μ	Dév. std. σ
HyVS	Err	0,196	0,18564	0,118	0,19903
	Tm	105,6 sec	34,8697	66,2 sec	21,7138
HyVC	Err	0,069	0,15953	0,059	0,19193
	Tm	90,3 sec	22,5196	86,6 sec	23,3486
HyVSC	Err	0,333	0,23210	0,236	0,23966
	Tm	138,4 sec	49,5880	111,4 sec	26,6831

Avant d'appliquer divers tests statistiques comparatifs à ces résultats, il nous semble utile de commenter ces résultats bruts. Tout d'abord, les tâches à réaliser ne semblaient pas être triviales pour les sujets du test. En effet, le temps moyen nécessaire pour les effectuer correctement varie d'environ une minute (66,2 sec) à plus de deux minutes (138,4 sec). En second lieu, on remarque que si les taux d'erreur sont faibles pour les tâches de type HyVC (~6 %) et acceptables pour les tâches de type HyVS avec des ellimaps (~12 %), ils posent question pour les tâches HyVS avec *treemaps* (~20%) et HyVSC (~24 %) avec ellimaps et deviennent inacceptables pour les tâches HyVSC (~ 33%) avec *treemaps*. Il est normal d'observer les taux d'erreurs les plus élevés et les temps de réponse les plus longs pour les tâches les plus complexes. Sur base de nos observations pendant les séances de tests, nous émettons l'hypothèse (sans disposer des informations nécessaires la valider formellement)

selon laquelle une partie significative des sujets ont éprouvé des difficultés à comprendre les questions les plus complexes ou dans certains cas à garder la concentration nécessaire pour y répondre correctement. Ceci devrait néanmoins faire l'objet d'une étude expérimentale spécifique.

b. Questions méthodologiques

En termes statistiques, nous devons évaluer si les moyennes de deux échantillons non indépendants sont significativement différentes. Dagnelie (2006, p. 228-235) propose différentes méthodes pour atteindre cet objectif. Il explique que *'d'une façon générale, les tests relatifs aux échantillons non indépendants, ou associés par paires ou par couples, sont basés sur le calcul des différences entre les couples d'observations.'*

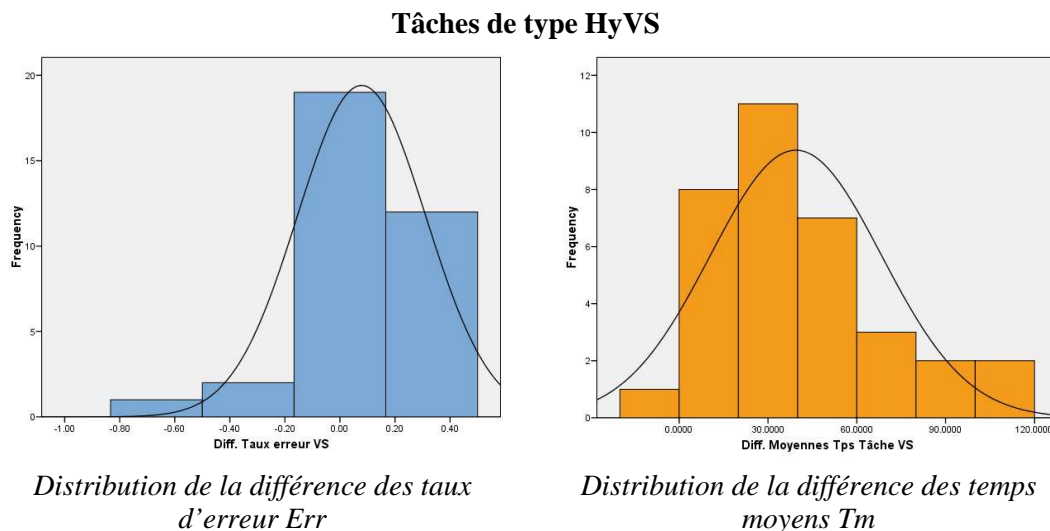
Nous pouvons donc dériver de nos résultats bruts la Table 36.

Table 36: Résultats présentés sous forme de différence des moyennes

		Différence des moyennes (Treemap – Ellimap)	
		Moyenne μ	Dév. std. σ
HyVS	Err	0,0784	0,23296
	Tm	39,4265 sec	28,9309
HyVC	Err	0,0098	0,1922
	Tm	3,6465 sec	24,0753
HyVSC	Err	0,098	0,2786
	Tm	27,0152 sec	43,8161

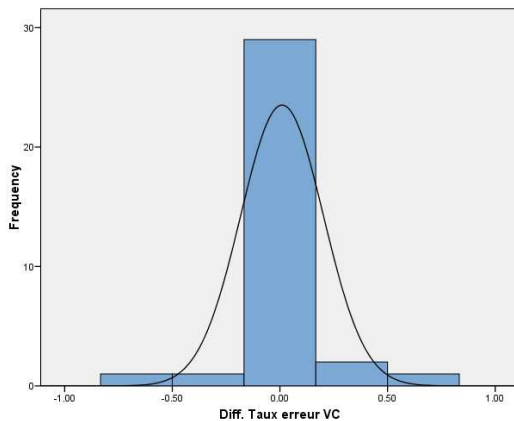
Les valeurs⁴⁷ de la moyenne et de la variance des observations sont évidemment utiles mais il est également important d'examiner les histogrammes de distribution des valeurs pour bien comprendre les résultats. Nous les présentons donc dans la Table 37.

Table 37: Histogrammes des distributions des résultats

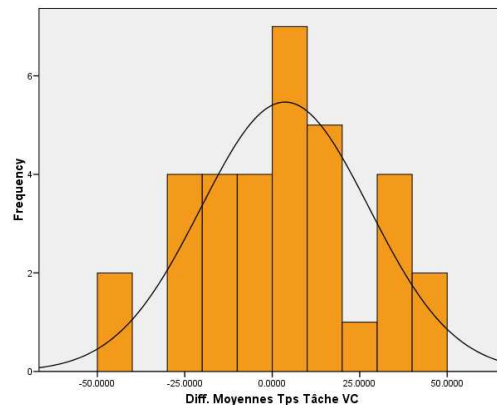


⁴⁷ Les calculs statistiques et les graphes présentés dans ce chapitre ont été réalisés avec le logiciel SPSS 15.0.

Tâches de type HyVC

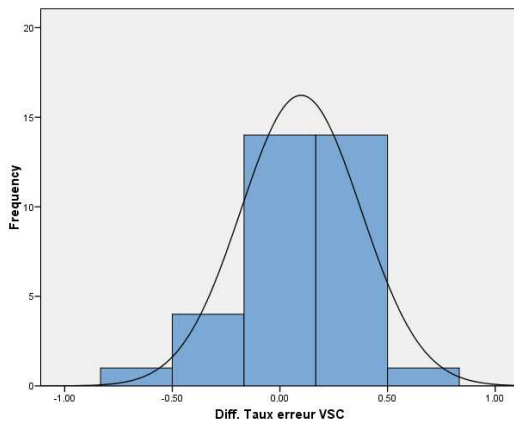


Distribution de la différence des taux d'erreur Err

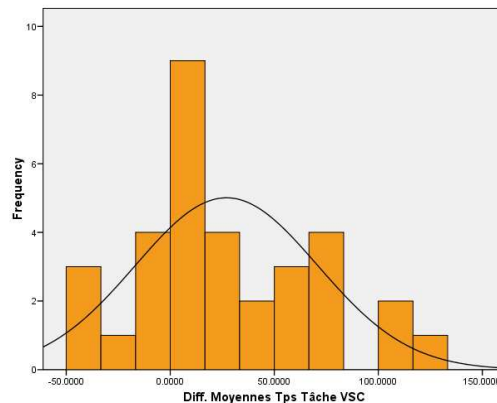


Distribution de la différence des temps moyens Tm

Tâches de type HyVSC



Distribution de la différence des taux d'erreur Err



Distribution de la différence des temps moyens Tm

Avant d'appliquer l'un ou l'autre test statistique à ces données, nous devons étudier la conformité des observations avec la loi de distribution sous-jacente à ces tests. En théorie statistique, cela correspond à réaliser un test d'ajustement (*goodness-of-fit test*) qui a pour but de savoir si 'un ensemble d'observations peut être considéré comme provenant d'une population d'un type donné' (cf. Dagnelie 2006, p. 64). Cette question est essentielle car si on veut ensuite analyser les résultats avec des tests statistiques paramétriques, cela exige de connaître quelle loi de distribution s'applique sur la population. Si nous ne pouvons pas identifier de loi de distribution applicable, nous devons utiliser des tests non-paramétriques. Rappelons que l'avantage des méthodes d'inférence statistique non paramétriques réside dans le fait qu'elles ne sont 'basées sur aucune hypothèse particulière relative aux distributions des populations-parents.' Par contre, 'elles sont moins efficaces ou moins puissantes que les méthodes paramétriques correspondantes [...]' (Dagnelie, 2006, p. 38).

Compte tenu de son importance dans les tests statistiques habituels, nous testerons si les observations peuvent être considérées comme issues d'une population normale (test de normalité). Dans notre contexte, cette opération peut être réalisée au moyen du test de Kolmogorov-Smirnov. Nous l'appliquerons pour déterminer si les observations sont

assimilables à une distribution normale dont la moyenne et la déviation standard sont celles calculées à partir de l'échantillon (cf. Table 36).

Sauf indication contraire, nous adopterons comme la valeur habituelle $p = 0,05$ comme limite pour accepter ou refuser l'hypothèse nulle dans nos tests statistiques.

Table 38: Test de normalité de la différence des taux d'erreur

	Différence des taux d'erreur Err		
	Tâches HyVS	Tâches HyVC	Tâches HyVSC
Nombre d'observations	34	34	34
Z de Kolmogorov-Smirnov	1,632	2,520	1,411
p	0,010	0,000	0,037
Conclusion	Hypothèse de normalité refusée	Hypothèse de normalité refusée	Hypothèse de normalité refusée

Table 39: Test de normalité de la différence des temps moyens

	Différence des temps moyens Tm		
	Tâches HyVS	Tâches HyVC	Tâches HyVSC
Nombre d'observations	34	33	33
Z de Kolmogorov-Smirnov	0,795	0,465	0,890
p	0,553	0,982	0,407
Conclusion	Hypothèse de normalité ne peut être refusée	Hypothèse de normalité ne peut être refusée	Hypothèse de normalité ne peut être refusée

Les tests de normalité nous indiquent donc que les différences de taux d'erreur ne peuvent pas être considérées comme issues d'une population normale de moyenne et variance égales à celles des observations. Par contre, les différences des temps moyens peuvent l'être.

Si les observations ne sont pas normalement distribuées, nous devons évaluer l'hypothèse d'égalité des moyennes (ou de nullité de la différence des moyennes) au moyen d'un test non paramétrique. Dans ce contexte, Dagnelie (2006, p. 232-234) explique que '*le test non paramétrique le plus courant, en ce qui concerne la comparaison de deux échantillons non indépendants, dans le cas de données continues, est [...] un test des rangs, dû à Wilcoxon (signed rank test ou Wilcoxon's matched pair signed rank test)*'.

Lorsque les observations ne sont pas distribuées selon une loi normale, il est cependant admis, sous certaines conditions, d'utiliser la méthode de test paramétrique du t-test par paires (*paired samples t-test*). Dagnelie (2006, p. 229) explique que '*cette méthode nécessite que les n couples d'observations constituent un échantillon aléatoire et simple, et que la population des différences soit normale. Cette dernière condition est cependant d'importance très secondaire, sauf pour des effectifs très réduits.*' En effet, '*en raison de la rapide convergence des distributions d'échantillonnage de la moyenne vers les distributions normales, la condition de normalité est toutefois très peu restrictive ici. Ce n'est que pour des effectifs très limités (distributions t à moins de 10 degrés de liberté) que cette condition a une réelle importance.*' De plus, '*d'une manière générale, les données étudiées peuvent être non*

seulement de nature continue, même fortement arrondies, mais aussi éventuellement de nature discontinue.' (Dagnelie, 2006, p. 208).

Pour résumer, le test de Wilcoxon est rigoureusement applicable tant aux différences des taux d'erreur qu'aux différences des temps moyens. Le t-test de Student est rigoureusement applicable aux différences des temps moyens et il est admis de l'appliquer aux différences des taux d'erreur. Par conséquent, nous appliquerons ces deux types de tests sur nos observations statistiques. Nous comparerons les résultats pour voir s'ils produisent des conclusions similaires.

c. Tests statistiques

Avant de détailler les résultats, il peut être utile de rappeler les hypothèses⁴⁸ que nous évaluons au moyen des tests de Wilcoxon et de Student.

- H1* : L'exactitude de la perception de la structure d'une hiérarchie est égale dans la représentation par *treemap* et par ellimap (cf. tâche HyVS).
- H2* : Le temps nécessaire pour percevoir correctement la structure d'une hiérarchie est égal dans la représentation par *treemap* et par ellimap (cf. tâche HyVS).
- H3* : L'exactitude de la perception du poids des nœuds d'une hiérarchie est égale dans la représentation par *treemap* et par ellimap (cf. tâche HyVC).
- H4* : Le temps nécessaire pour percevoir correctement le poids des nœuds d'une hiérarchie est égal dans la représentation par *treemap* et par ellimap (cf. tâche HyVC).
- H5* : La qualité de perception combinée de la structure et du poids des nœuds d'une hiérarchie est égale dans la représentation par *treemap* et par ellimap (cf. tâche HyVSC).
- H6* : Le temps nécessaire à une perception combinée correcte de la structure et du poids des nœuds d'une hiérarchie est égal dans la représentation par *treemap* et par ellimap (cf. tâche HyVSC).

Nous présenterons les résultats en les regroupant par type de tâche. Pour chacun d'entre eux, nous expliquerons d'abord les conclusions du test non-paramétrique de Wilcoxon puis celles du t-test de Student.

Pour les tâches liées à la **perception de la structure** (HyVS), le test de Wilcoxon nous indique que l'hypothèse H1* d'égalité des taux d'erreur ne peut être rejetée à un seuil de signification de 5%. Néanmoins, la valeur obtenue (5,9%) est très proche du seuil de rejet et nous tiendrons compte de cet élément dans nos conclusions. Rigoureusement, ce résultat signifie qu'en rejetant l'hypothèse nulle, nous avons une probabilité de 5,9% de nous tromper.

En ce qui concerne l'égalité des temps moyens (H2*), le test de Wilcoxon permet de rejeter clairement l'hypothèse d'égalité.

Table 40: Test de Wilcoxon pour les tâches de type HyVS

			Wilcoxon rank test
HyVS	H1*	Err	$Z = -1,886$ $p = 0,059$
	⇒ hyp. d'égalité des taux d'erreur ne peut être rejetée mais résultats proches du rejet		
	H2*	Tm	$Z = -5,044$ $p < 0,001$
	⇒ hyp. d'égalité des temps moyens peut être rejetée		

⁴⁸ Les hypothèses Hi* correspondent aux hypothèses Hi dans lesquelles le terme '*diffère*' est remplacé par '*est égal*'. Cette adaptation est justifiée par le fait que les tests statistiques utilisés évaluent l'égalité de moyennes.

Si nous appliquons le t-test de Student sur ces données, nous obtenons les résultats présentés dans la Table 41.

Table 41: t-Test de Student pour les tâches de type HyVS

			t- test
HyVS	H1*	Err	t = 1,963 degrés de liberté: 33 p = 0,058 intervalle de confiance à 95%: [-0,0029 , 0,1597]
	⇒ hyp. d'égalité des taux d'erreur ne peut être rejetée mais résultats proches du rejet		
	H2*	Tm	t = 7,946 degrés de liberté: 33 p < 0,000 intervalle de confiance à 95%: [29,332 , 49,521]
	⇒ hyp. d'égalité des temps moyens peut être rejetée		

En conclusion, il existe des indications tendant à mettre en évidence des taux d'erreurs inférieurs pour les ellimaps que pour les *treemaps* pour les tâches HyVS. Cependant, ces indications ne peuvent à ce stade être considérées comme des preuves irréfutables.

Par contre, au niveau du temps moyen pour réaliser correctement une tâche liée à la perception de la structure, une différence est clairement mise en évidence entre les deux techniques de visualisation. Pour ce type de tâche, la différence moyenne d'environ 39 secondes correspond en termes relatifs à un temps supérieur de 59,6 % pour le *treemap* par rapport à ce que nécessite l'ellimap.

Au niveau des tâches relatives à la **perception du poids des nœuds** (HyVC), le test de Wilcoxon montre que ni l'hypothèse d'égalité des taux d'erreur (H3*), ni l'hypothèse d'égalité des temps moyens (H4*) ne peuvent être rejetées.

Table 42: Test de Wilcoxon pour les tâches de type HyVC

			Wilcoxon rank test
HyVC	H3*	Err	Z = -0,276 p = 0,783
	⇒ hyp. d'égalité des taux d'erreur ne peut être rejetée		
	H4*	Tm	Z = -0,715 p = 0,475
	⇒ hyp. d'égalité des temps moyens ne peut être rejetée		

Le t-test de Student fournit des conclusions similaires (cf. Table 43).

Table 43: t-Test de Student pour les tâches de type HyVC

			t- test	
HyVC	H3*	Err	t = 0,297 p = 0,768 intervalle de confiance à 95%: [-0,0573 , 0,0769]	degrés de liberté: 33
	⇒ hyp. d'égalité des taux d'erreur ne peut être rejetée			
	H4*	Tm ⁴⁹	t = 0,870 p = 0,391 intervalle de confiance à 95%: [-4,8903 , 12,1832]	degrés de liberté: 32
	⇒ hyp. d'égalité des temps moyens ne peut être rejetée			

Pour la perception du poids des nœuds, les deux techniques de visualisation ne sont donc pas significativement différentes. On note d'ailleurs que la différence moyenne entre les temps nécessaires pour accomplir ce type de tâche n'est que d'environ 3 secondes, ce qui ne représente pas un écart relatif important par rapport aux temps moyens (~90 sec).

En ce qui concerne les tâches plus complexes de **perception combinée de la structure et du poids des nœuds**, le test de Wilcoxon indique que l'hypothèse H5* d'égalité des taux d'erreur ne peut être rejetée mais que si nous acceptons une probabilité d'erreur de 5,2 % au lieu de 5% elle le serait. Nous sommes donc en face d'une situation où un certain degré d'appréciation doit jouer un rôle. Il est par contre très clair que l'hypothèse H6* d'égalité des temps moyens peut être rejetée.

Table 44: Test de Wilcoxon pour les tâches de type HyVSC

			Wilcoxon rank test	
HyVSC	H5*	Err	Z = -1,945	p = 0,052
	⇒ hyp. d'égalité des taux d'erreur ne peut être rejetée mais résultats très proches du rejet			
	H6*	Tm	Z = -3,189	p = 0,001
	⇒ hyp. d'égalité des temps moyens peut être rejetée			

En appliquant le t-test de Student, nous obtenons les résultats de la Table 45.

Ce test confirme que les taux d'erreur associés aux tâches de type HyVSC sont différents pour les deux techniques de visualisation, à un seuil de signification proche de 0,05 (cf. résultats du test de Wilcoxon).

⁴⁹ Nous avons exclu du calcul des temps moyens les cas où la différence n'était pas calculable, parce que le temps moyen pour la condition 'treemap' ou 'ellimap' était manquant.

Table 45: t-Test de Student pour les tâches de type HyVSC

			t- test
HyVSC	H5*	Err	t = 2,052 degrés de liberté: 33 p = 0,048 intervalle de confiance à 95%: [0,0008 , 0,1953]
	⇒ hyp. d'égalité des taux d'erreur peut être rejetée		
	H6*	Tm ⁴⁹	t = 3,542 degrés de liberté: 32 p = 0,001 intervalle de confiance à 95%: [11,4786 , 42,5517]
	⇒ hyp. d'égalité des temps moyens peut être rejetée		

Au niveau des taux d'erreur, nous observons une différence entre les *treemaps* et les *ellimaps* à un seuil de signification de ~6%. Bien que ce seuil soit (légèrement) supérieur à ce qui est communément admis, nous considérons cependant que nos résultats plaident en faveur d'une différence sans toutefois en constituer une preuve.

La situation est plus claire en ce qui concerne les temps moyens. La différence moyenne d'environ 27 secondes correspond en termes relatifs à un temps supérieur de 24 % pour le *treemap* pour une tâche de perception combinée de la structure et du poids des nœuds.

4.3.2. Evaluation subjective

L'évaluation subjective des deux techniques de visualisation se base sur un questionnaire de satisfaction (cf. p. 271).

Nous avons codé les résultats sur une échelle allant de 1 (*très mauvais*) à 5 (*très bon*). Les résultats sont présentés dans la Table 46. Les moyennes représentent la note moyenne obtenue pour l'avis subjectif des sujets.

Table 46: Résultats bruts du questionnaire de satisfaction

Item	Treemap		Ellimap	
	Moyenne μ	Dév. std. σ	Moyenne μ	Dév. std. σ
Opinion globale	2,53	0,825	4,26	0,666
Opinion sur la perception de la structure des données (HyVS)	2,62	0,985	4,35	0,691
Opinion sur la perception du poids des nœuds (HyVC)	2,88	1,094	4,09	0,621

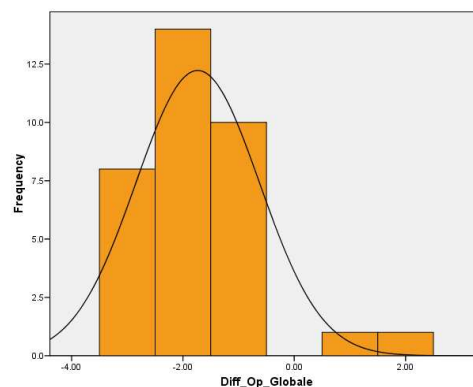
Nous présentons ces résultats sous la forme de différences dans la Table 47.

Table 47: Résultats du test de satisfaction exprimés sous forme de différences

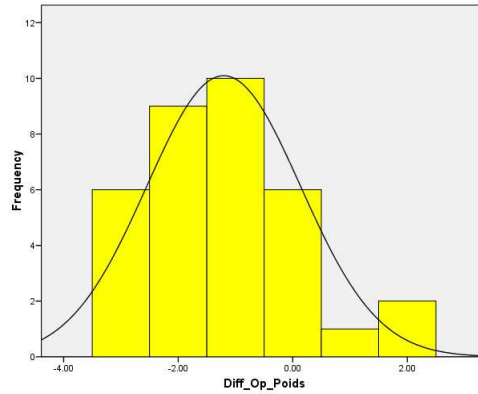
	Différence des moyennes (Treemap – Ellimap)	
	Moyenne μ	Dév. std. σ
Opinion globale	-1,7353	1,1094
Opinion sur la perception de la structure des données (HyVS)	-1,7353	1,2627
Opinion sur la perception du poids des nœuds (HyVC)	-1,2059	1,3434

Comme pour les taux d'erreur et les temps moyens nécessaires pour accomplir une tâche, nous fournissons également les distributions des différences entre les valeurs obtenues pour les techniques *treemap* et *ellimap* (cf. Table 48).

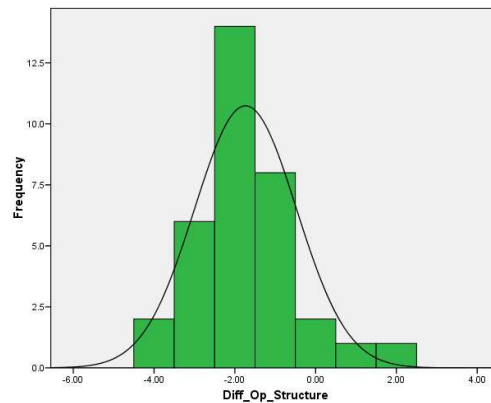
Table 48: Histogrammes de distribution des résultats du questionnaire de satisfaction



Distribution de la différence d'opinion globale entre treemap et ellimap



Distribution de la différence d'opinion sur la perception du poids des nœuds



Distribution de la différence d'opinion sur la perception de la structure des données

Nous avons appliqué un test de normalité de Kolmogorov-Smirnov pour voir quels tests de comparaison des moyennes étaient applicables.

Table 49: Test de normalité sur les différences d'opinions subjectives

	Différence des opinions subjectives		
	Global	Poids des nœuds	Structure des données
Nombre d'observations	34	34	34
Z de Kolmogorov-Smirnov	1,407	1,017	1,342
p	0,038	0,252	0,055
Conclusion	Hypothèse de normalité refusée	Hypothèse de normalité ne peut être refusée	Hypothèse de normalité ne peut être refusée

Pour des raisons similaires aux cas des différences des taux d'erreur et des temps moyens, nous avons décidé d'appliquer à la fois le test non-paramétrique de Wilcoxon et le t-test paramétrique de Student pour déterminer si les opinions subjectives collectées via le questionnaire de satisfaction diffèrent pour les deux techniques de visualisation.

Table 50: Test de Wilcoxon sur la différence d'opinion sur les deux techniques

	Wilcoxon rank test	
	Z	p
Opinion globale	- 4,734	0,000
	⇒ hyp. d'égalité des opinions globales peut être rejetée	
Opinion sur la perception du poids des nœuds (HyVC)	- 3,770	0,000
	⇒ hyp. d'égalité des opinions sur la perception du poids des nœuds peut être rejetée	
Opinion sur la perception de la structure des données (HyVS)	- 4,598	0,000
	⇒ hyp. d'égalité des opinions sur la perception de la structure des données peut être rejetée	

Selon le test de Wilcoxon, les différentes opinions subjectives sont clairement différentes pour le *treemap* et pour l'ellimap, en faveur de l'ellimap.

Table 51: t-Test de Student sur la différence d'opinion sur les deux techniques

	t- test	
	t	degrés de liberté
Opinion globale	t = - 9,120 p = 0,000 intervalle de confiance à 95% : [- 2,1224 , - 1,3482]	33
	⇒ hyp. d'égalité des opinions globales peut être rejetée	
Opinion sur la perception du poids des nœuds (HyVC)	t = - 5,234 p = 0,000 intervalle de confiance à 95% : [- 1,6746 , - 0,7371]	33
	⇒ hyp. d'égalité des opinions sur la perception du poids des nœuds peut être rejetée	
Opinion sur la perception de la structure des données (HyVS)	t = - 8,013 p = 0,000 intervalle de confiance à 95% : [- 2,1759 , - 1,2947]	33
	⇒ hyp. d'égalité des opinions sur la perception de la structure des données peut être rejetée	

Le t-test de Student confirme les conclusions du test de Wilcoxon.

En conclusion, il apparaît donc que l'ellimap récolte une meilleure note que le *treemap* en ce qui concerne sa perception subjective globale. Alors que le *treemap* est jugé en moyenne 'moyen à mauvais' (2,53), l'ellimap est jugé 'bonne à très bonne' (4,26). Au niveau des tâches, l'ellimap est aussi mieux appréciée tant pour les tâches de perception de la structure que de celle de perception du poids des nœuds.

Ces résultats ne sont pas très surprenants car ils confirment d'autres études. Par exemple, lors de sa comparaison de quatre navigateurs hiérarchiques, Andrews (2007) a montré que les

utilisateurs notaient subjectivement moins bien le navigateur *treemap* que les autres navigateurs pour une variété de facteurs.

Enfin, il est intéressant de noter que les mesures objectives (Err, Tm) et les opinions subjectives concordent. Alors que dans certaines expériences, les mesures contredisent la perception des individus, nous n’observons pas ce phénomène dans notre étude.

4.3.3. Conclusions

Au moment de tirer des conclusions sur notre évaluation comparative, il peut être utile de rappeler la motivation initiale qui nous a conduits à imaginer la technique des ellimaps. Sur base de la littérature et de notre réflexion personnelle, nous avons constaté que les *treemaps* étaient performants pour visualiser le poids relatif des nœuds dans une structure hiérarchique. Par contre, ils présentaient des faiblesses en ce qui concerne la perception de cette structure.

Nous avons imaginé la technique des ellimaps pour tenter de répondre à cette limitation. Notre objectif consistait donc à améliorer la perception de la structure du jeu de données sans (trop) pénaliser celle du poids relatif des nœuds.

L’évaluation comparative décrite précédemment visait à valider ou invalider notre démarche théorique par des données empiriques.

D’un point de vue général, les conclusions sont globalement positives. Elles confirment nos intuitions initiales. En effet, dans les conditions expérimentales décrites, il apparaît que, comparée au *treemap*, l’ellimap améliore la perception de la structure des données hiérarchiques sans pénaliser la perception du poids des nœuds.

Pour résumer, nous pouvons tirer de notre évaluation les enseignements suivants.

- Les ellimaps et les *treemaps* conduisent à des taux d’erreur et à des temps moyens de réponse correcte similaires pour les tâches de perception du poids des nœuds.
- Une tendance est mise en évidence pour supporter l’assertion selon laquelle les ellimaps conduisent à des taux d’erreur inférieurs à ceux des *treemaps* pour les tâches liées à la perception de la structure. Cependant, nos observations conduisent à des valeurs proches de la limite d’acceptation / de rejet des tests statistiques.
- Pour les tâches liées à la perception de la structure des données, les ellimaps permettent de répondre correctement de manière plus rapide que les *treemaps*.
- Au niveau des tâches plus complexes qui nécessitent la perception de la structure des données et du poids des nœuds, nous observons une tendance qui nous incite à penser que les taux d’erreur sont plus faibles avec les ellimaps qu’avec les *treemaps*. Cependant, la rigueur statistique nous interdit de conclure fermement sur ce point.
- Les ellimaps permettent de répondre plus vite de manière correcte que les *treemaps* à des questions portant sur la perception combinée de la structure des données et du poids des nœuds.

Au niveau méthodologique, nous attirons l’attention sur le fait que nos résultats sont très focalisés car ils ne concernent que la disposition spatiale des deux techniques de visualisation. Fondamentalement, nous avons montré qu’un utilisateur appartenant à notre population cible perçoit plus facilement la structure d’une hiérarchie de taille moyenne dans la vue ellimap que dans la vue *treemap*. De tels résultats n’auraient probablement pas pu être obtenus si nous avions comparé des implémentations disposant de techniques d’interaction avancées (p.ex. vues couplées, fonctionnalité de recherche de nœud) car celles-ci peuvent compenser certaines limitations du concept de base. Globalement, ces résultats nous confortent dans l’idée que l’approche ellimap mérite de continuer à être étudiée. Des améliorations sont sans doute possibles en adaptant certaines évolutions des *treemaps* au cas des ellimaps.

Chapitre 9 : Prototype 'Ellimaps'

« C'est par l'expérience que la science et l'art font leur progrès chez les hommes. »

Aristote

1. Fonctionnalités

1.1 Organisation générale

L'interface utilisateur principale se compose de deux parties (cf. Fig. 163). La zone supérieure regroupe les éléments de contrôle interactif de l'affichage et une zone d'affichage d'informations détaillées sur le nœud sélectionné.

La zone inférieure est dédiée à la visualisation des données proprement dite. Cette zone peut être scindée en deux parties permettant l'affichage simultané d'une vue en arbre (partie inférieure gauche) et d'une vue ellimap ou *treemap* (partie inférieure droite).

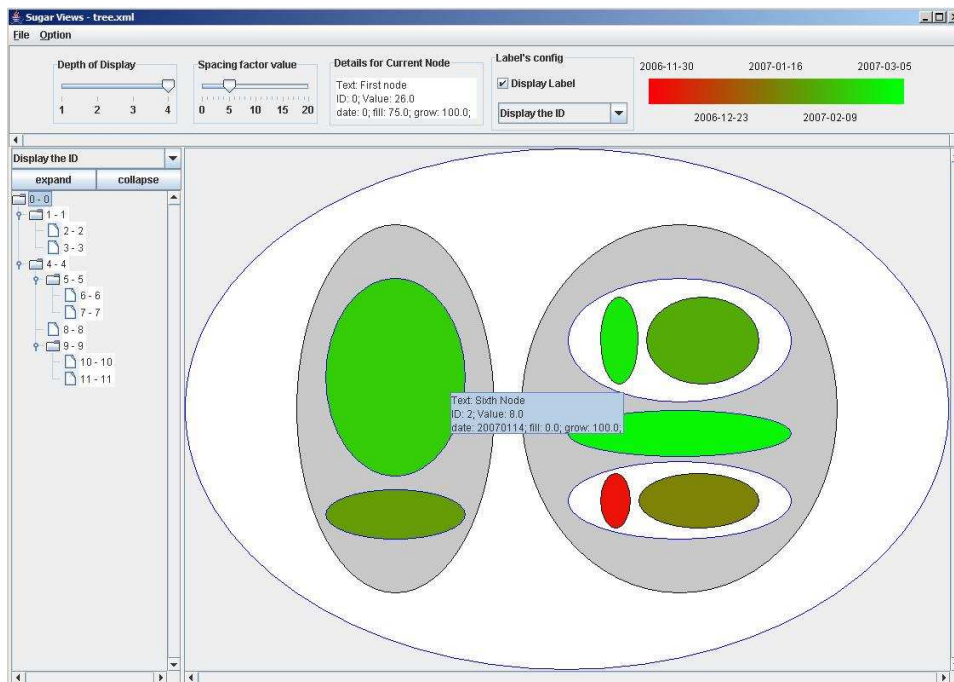


Fig. 163: Interface utilisateur principale du prototype

Les nombreuses options de configuration sont accessibles via une fenêtre dédiée (cf. Fig. 164) dans laquelle les catégories de paramètres sont regroupées dans des onglets spécifiques :

- mise en relation du fichier XML et des variables graphiques (→ 'XML') ;
- politique de coloriage (→ 'Color') ;
- configuration des étiquettes (→ 'Labels') ;
- paramétrage de l'algorithme de visualisation (→ 'Orientation') ;
- choix de l'algorithme de visualisation : ellimap / *treemap* (→ 'Display') ;
- configuration des ellimaps-réservoirs (→ 'Filling').



Fig. 164: Fenêtre de configuration

Nous détaillerons dans la suite de cette section ces différentes options qui rendent le prototype particulièrement générique.

1.2 Interaction avec les données

Dans cette section, nous expliquons les fonctionnalités essentiellement liées à l'interaction avec les données.

En premier lieu, le prototype permet de naviguer dans le jeu de données de différentes manières, au choix de l'utilisateur. Dans la vue en arbre, les fonctionnalités classiques sont supportées (p.ex. double clic pour faire apparaître les descendants directs d'un nœud). Dans la vue ellimap / *treemap*, un clic dans une ellipse / rectangle positionne le nœud associé comme racine visuelle de la représentation (navigation descendante). La navigation ascendante est réalisée via le clic droit. L'utilisateur peut soit monter d'un niveau dans la hiérarchie, soit revenir à la racine.

Si les vues en arbre et ellimap / *treemap* sont affichées simultanément, elles sont couplées. L'utilisateur peut choisir celle avec laquelle il préfère interagir à un moment donné. Le couplage dynamique des vues ne signifie pas que les informations visualisées dans celles-ci soient exactement identiques. En effet, la synchronisation porte sur le nœud actif sélectionné. Lorsque l'utilisateur clique dans un nœud dans la vue en arbre, la vue ellimap / *treemap* est adaptée pour que ce nœud devienne la racine visuelle de la représentation. Par contre, la profondeur d'affichage reste indépendante dans les deux vues, de même que les informations placées dans les étiquettes associées aux nœuds.

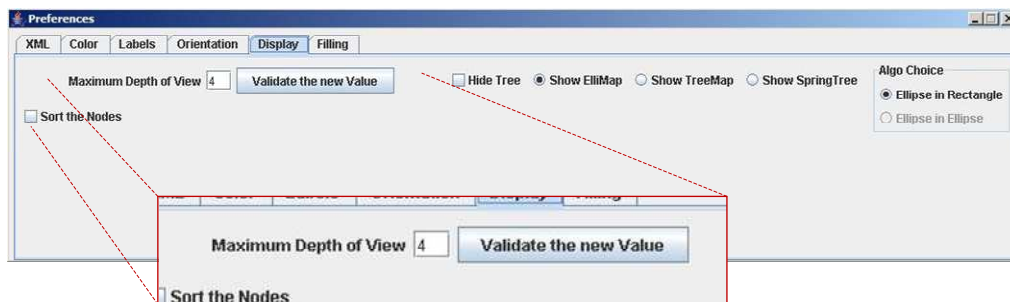


Fig. 165: Paramétrage du niveau maximum de profondeur d'affichage

Le choix de la profondeur d'affichage dans les vues ellimap / *treemap* est illustrée par la Fig. 166. En déplaçant un curseur, la vue ellimap s'adapte en temps réel selon le principe du '*query dynamique*'. Notons que le niveau de profondeur maximum affichable (c.-à.d. la gamme des valeurs possibles du curseur associé) est un paramètre éditable dans la fenêtre de configuration (cf. Fig. 165).

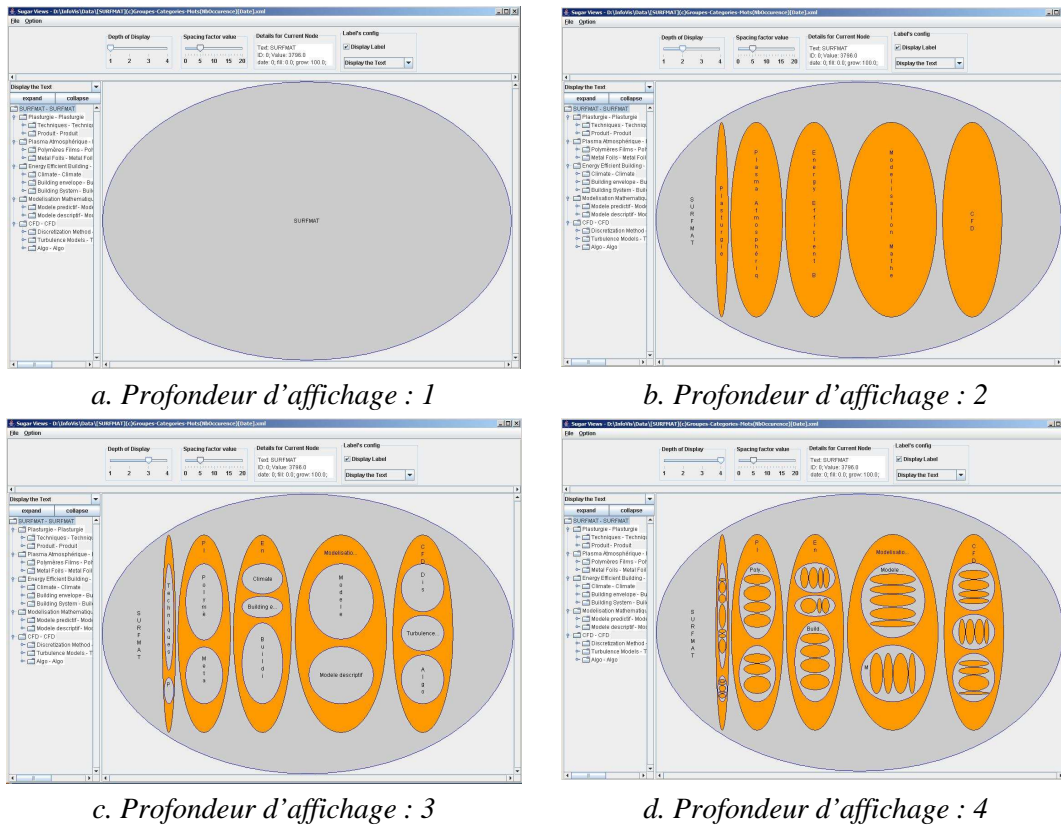


Fig. 166: Configuration de la profondeur d'affichage dans la vue ellimap.

Une autre fonctionnalité particulièrement utile consiste dans la possibilité offerte à l'utilisateur de choisir dynamiquement les informations visibles pour chaque nœud dans chacune des vues, de façon indépendante.

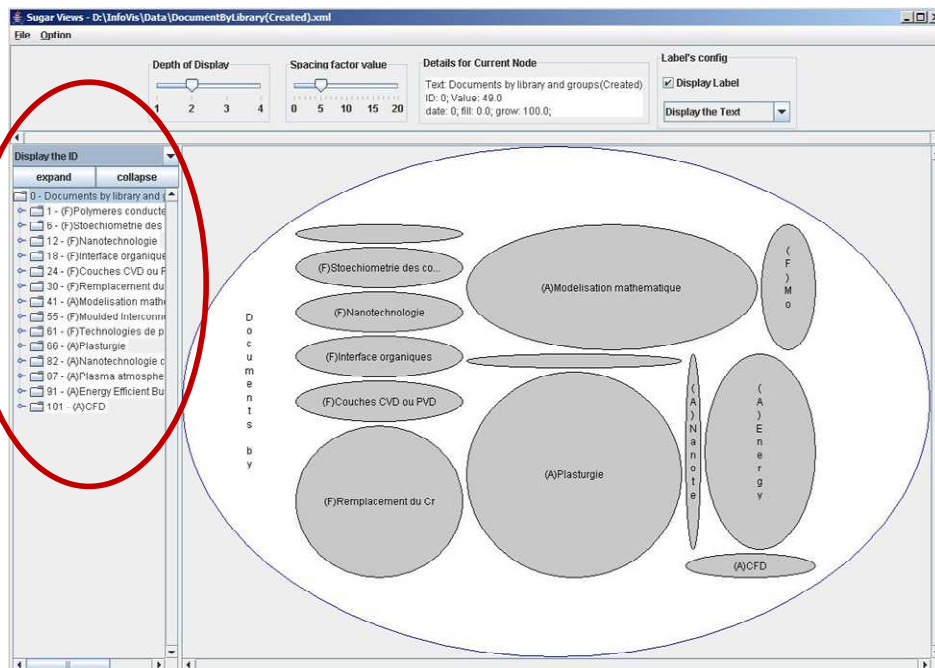


Fig. 167: Choix des informations affichées dans la vue en arbre

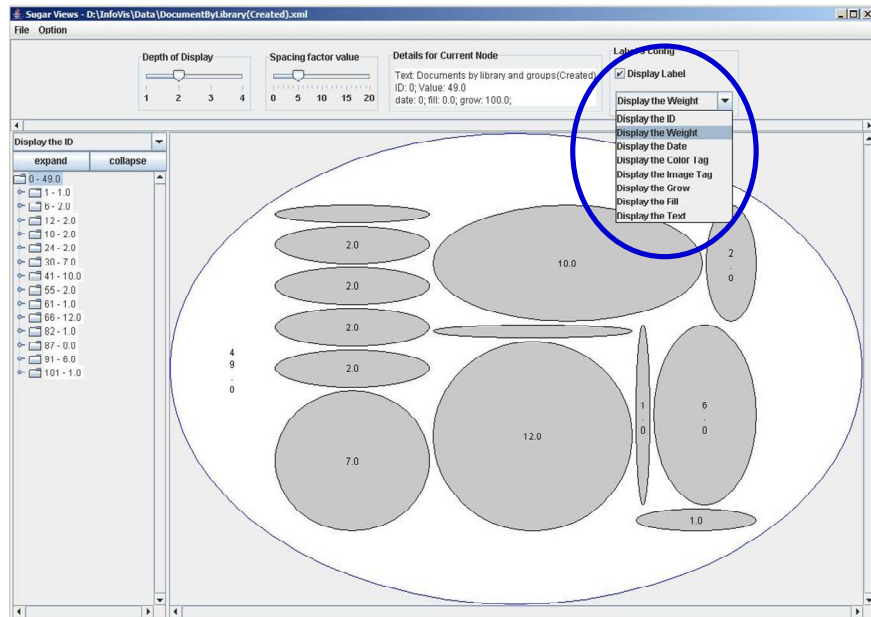


Fig. 168: Choix des informations affichées dans la vue ellimap

Par exemple, dans la Fig. 167, le nom des nœuds (propriété *'text'*) est affiché dans la vue ellimap et le couple *'identifiant – nom'* dans la vue en arbre. Par contre, dans la Fig. 168, le poids des nœuds (propriété *'weight'*) est affiché dans la vue ellimap et le couple *'identifiant – poids'* dans la vue en arbre.

Si l'aspect visuel de la représentation ellimap ou *treemap* n'est pas satisfaisant aux yeux de l'utilisateur, le facteur d'espacement entre les ellipses ou les rectangles (cf. p. 192) peut être modifié par un curseur sur le principe du '*query dynamique*' (cf. Fig. 169).

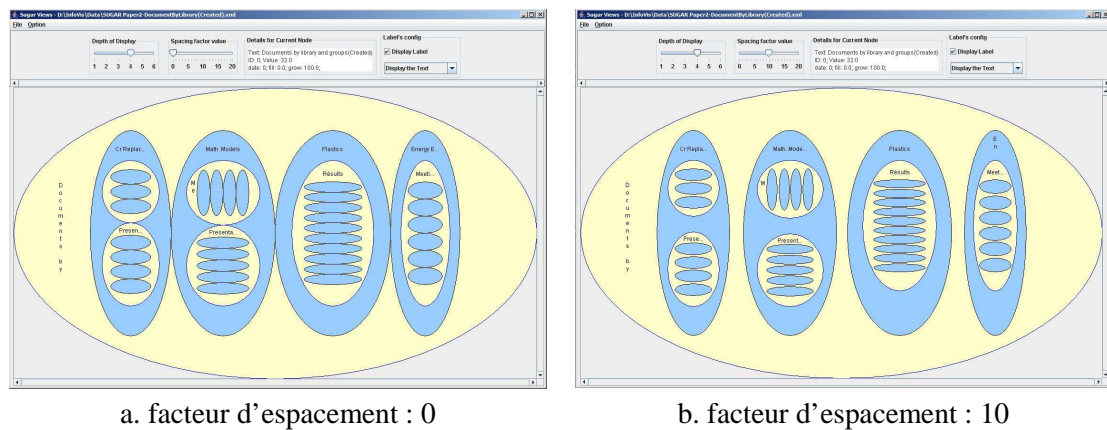


Fig. 169: Réglage interactif du facteur d'espacement

En ce qui concerne les fonctionnalités d'interaction avec la représentation graphique des données, il faut aussi mentionner la présence d'étiquettes dynamiques (*'tooltips'*) qui apparaissent quand le curseur de la souris est placé sur une ellipse ou un rectangle. Cette étiquette détaille les informations disponibles sur le nœud (cf. Fig. 170).

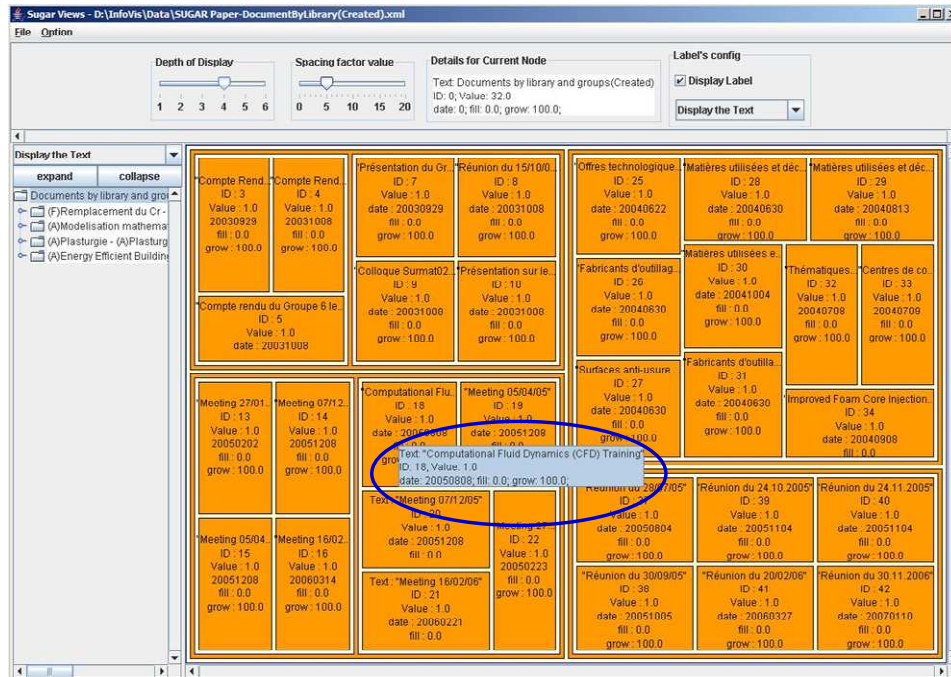


Fig. 170: Étiquettes dynamiques ('squarified treemap')

L'affichage des informations sur les nœuds peut être paramétré via la fenêtre de configuration (cf. Fig. 171). Il est possible de choisir la position à l'intérieur de l'ellipse ou du rectangle (gauche / droite, haut / bas), d'activer une fonction qui calcule l'espace disponible et tronque le texte à afficher en conséquence ou de supprimer tout affichage dans les ellipses ou les rectangles. Le niveau jusqu'auquel les étiquettes sont affichées dans les vues ellimap et *treemap* est également configurable.

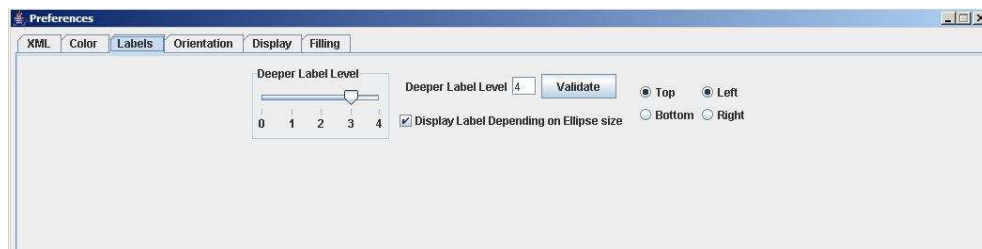


Fig. 171: Paramétrage des étiquettes

1.3 Algorithmes de positionnement des nœuds

Différents algorithmes de positionnement spatial des nœuds de la hiérarchie à visualiser ont été intégrés dans le prototype. Ils sont spécifiés dans la Table 52. Une fenêtre de configuration permet le choix de la technique de visualisation et de l'orientation (cf. Fig. 172).

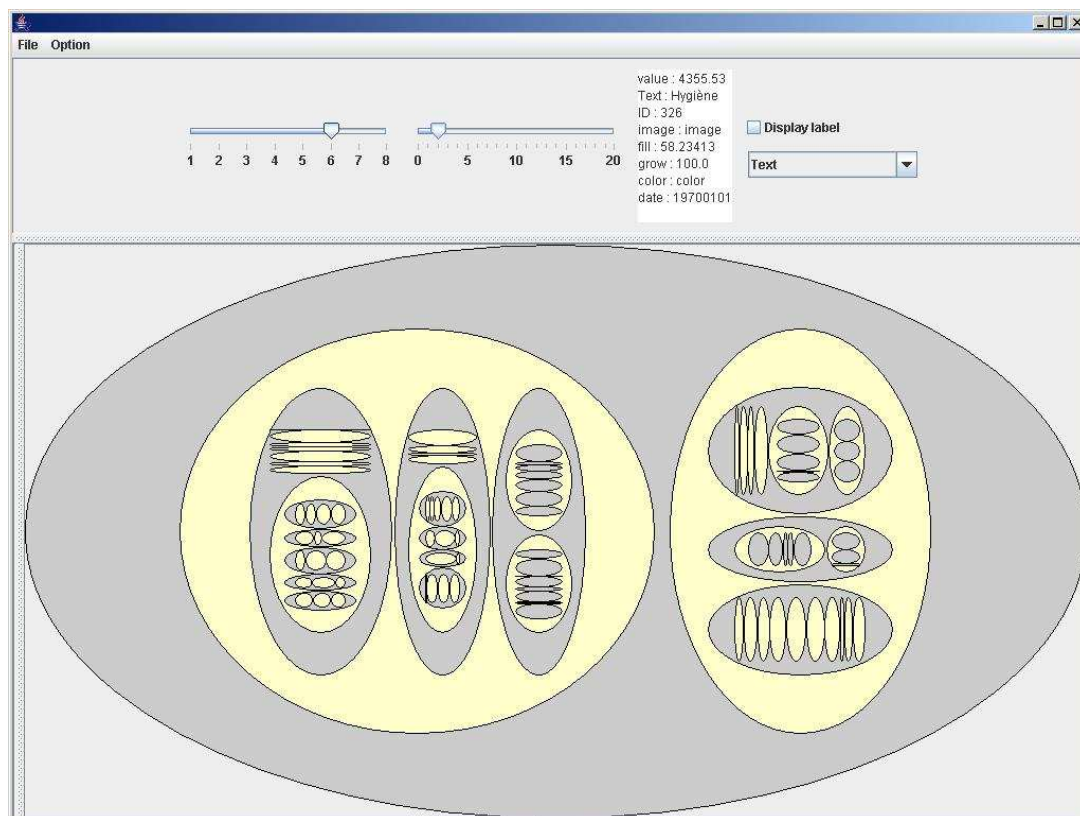


Fig. 172: Choix de l'orientation de la disposition des nœuds

Table 52: Algorithmes de positionnement disponibles dans le prototype

Technique	Variante
Treemap	<i>Division imposée selon l'axe horizontal</i>
	<i>Division imposée selon l'axe vertical</i>
	<i>Division alternée</i>
	<i>Division dépendante</i>
	<i>Squarified</i>
Ellimap	<i>Division imposée selon l'axe horizontal</i>
	<i>Division imposée selon l'axe vertical</i>
	<i>Division alternée</i>
	<i>Division dépendante</i>
	<i>Circularisée</i>
	<i>Algorithme DGC</i>

Les figures ci-dessous illustrent quelques-unes des possibilités pour visualiser un même jeu de données au moyen de variantes de la technique des ellimaps.

**Fig. 173: Ellimap avec algorithme de division dépendante**

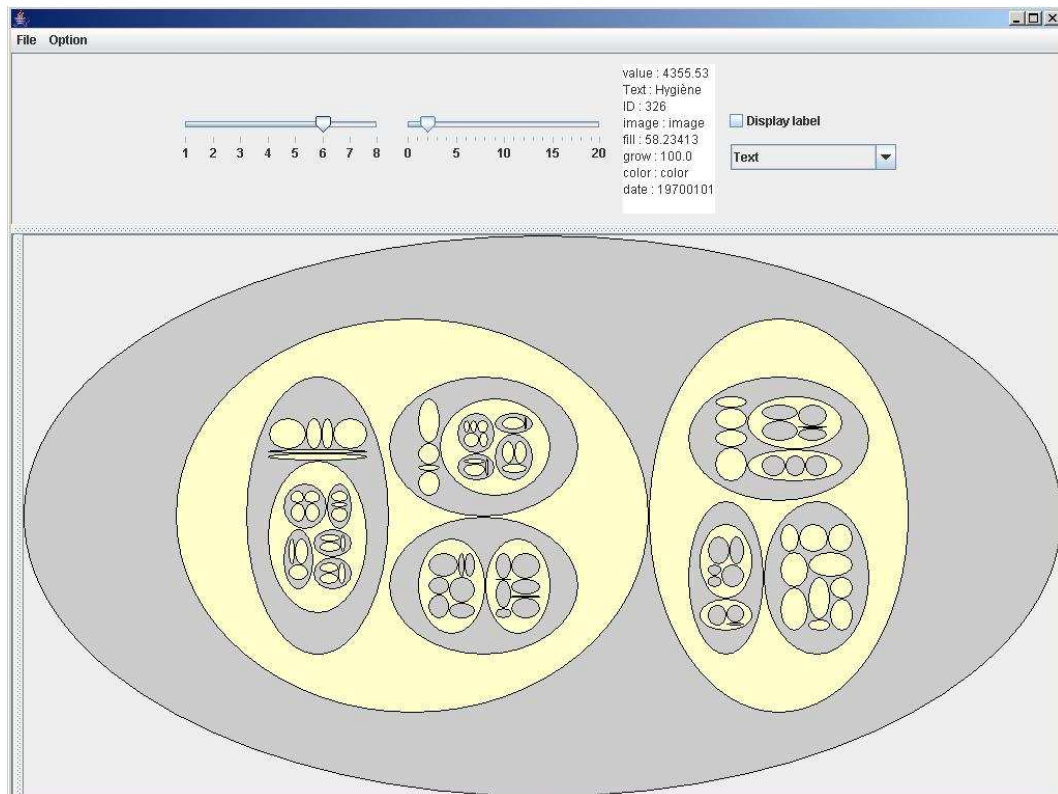


Fig. 174: Ellimap circularisée

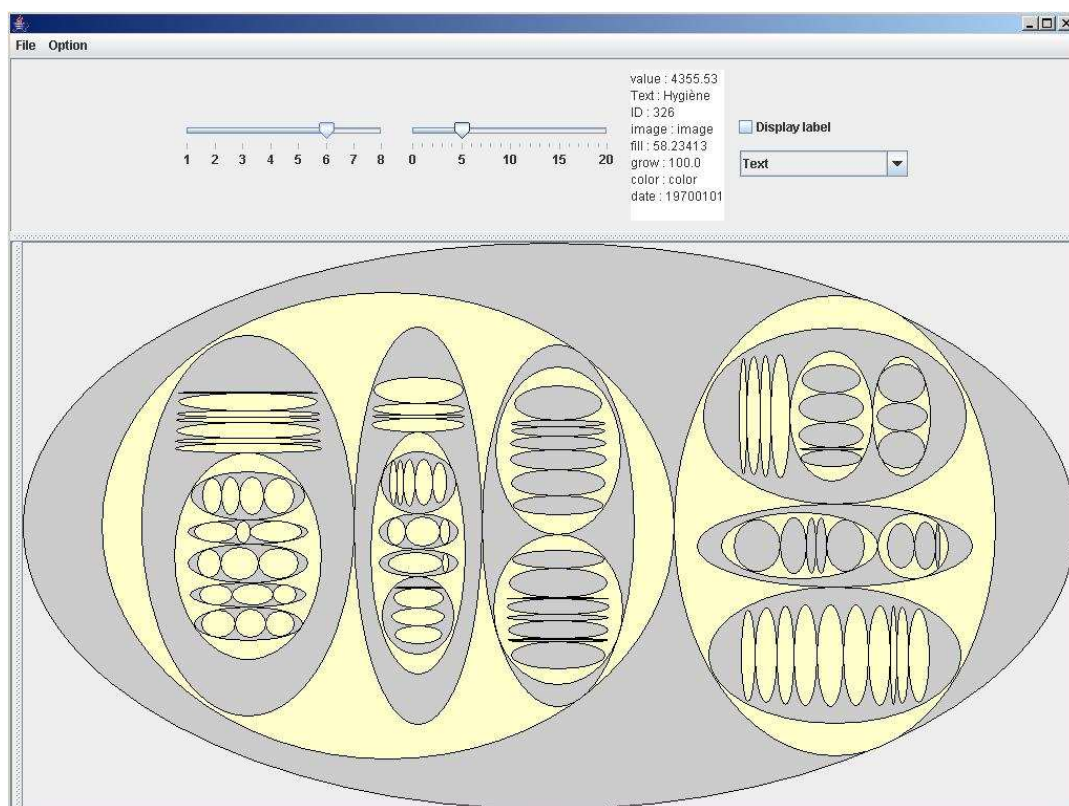


Fig. 175: Ellimap avec algorithme DGC

1.4 Tri

Le prototype intègre une fonction permettant de trier les nœuds selon leur poids. Par défaut, les nœuds sont affichés dans l'ordre selon lequel ils apparaissent dans les données brutes. Bien qu'elle ne soit pas optimale dans tous les cas d'un point de vue graphique, cette option permet de faciliter certaines opérations relatives à la pondération des nœuds.

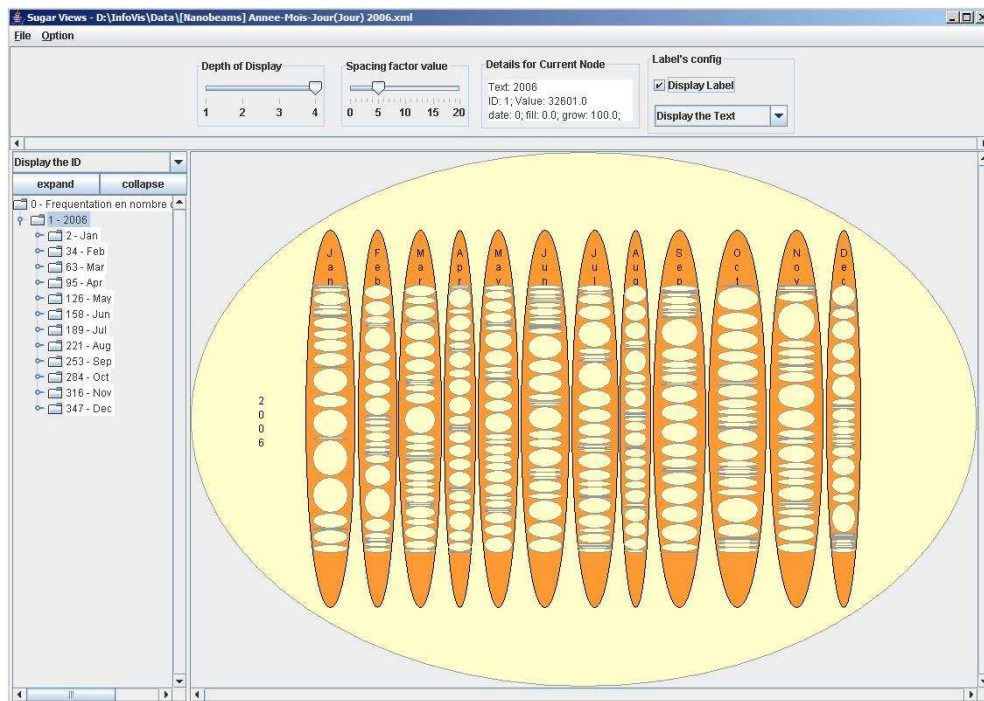


Fig. 176: Ellimap avec données non triées

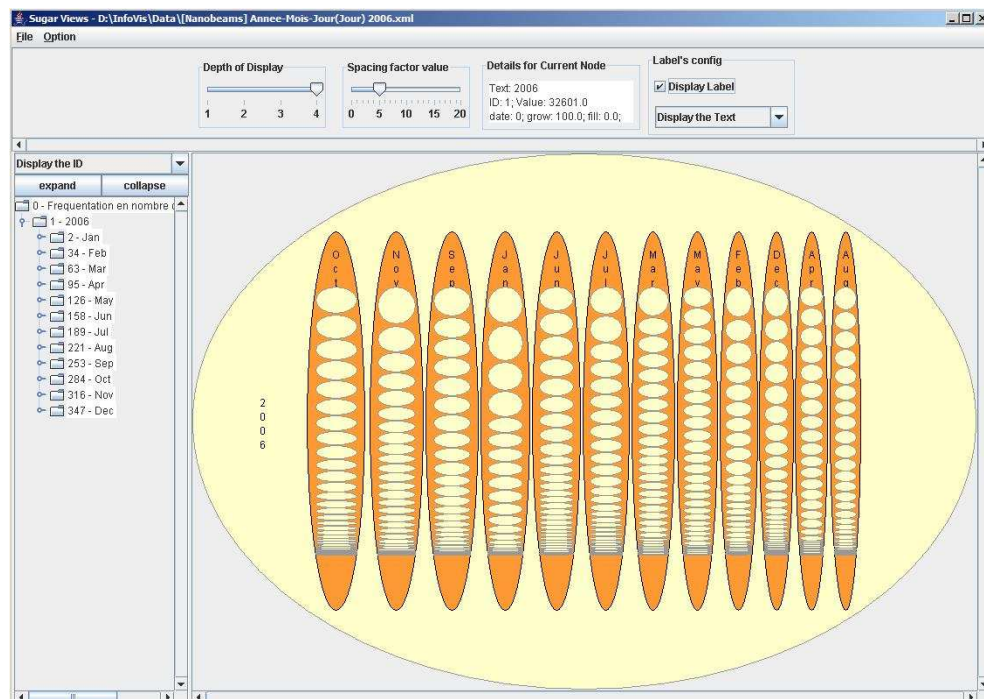


Fig. 177: Ellimap avec données triées en ordre descendant

1.5 Politique de coloriage

La couleur est une variable visuelle dont l’importance n’est plus à démontrer. Nous avons donc inclus dans le prototype un ensemble de possibilités de paramétrage pour le coloriage des ellipses et rectangles dans les vues ellimap et *treemap*. Différentes options sont disponibles.

La plus simple permet d’alterner le coloriage des niveaux hiérarchiques avec deux couleurs (cf. Fig. 178). La couleur du contour des formes imbriquées peut également être configurée de manière semblable. Le changement de couleur alterné permet d’accentuer la mise en évidence de la structure hiérarchique.

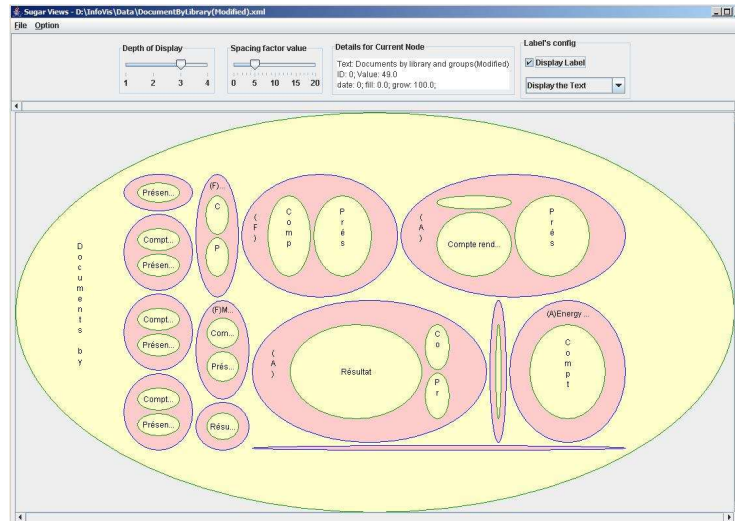


Fig. 178: Coloriage des ellipses en mode alterné

Une autre possibilité consiste à appliquer un pseudo-coloriage aux formes imbriquées. Dans ce cas, une échelle chromatique est mise en correspondance avec une propriété continue ordonnable des nœuds (p.ex. date de dernière modification d’un document, taux de croissance des visites d’une page web). Le prototype offre à l’utilisateur le choix entre deux échelles bi-chromatiques (‘rouge – vert’, cf. Fig. 180 et ‘vert – rouge’) et une échelle mono-chromatique (niveaux de vert, cf. Fig. 179). Selon la sémantique des données, l’utilisateur peut faire apparaître en rouge les valeurs faibles ou les valeurs élevées d’une propriété particulière.

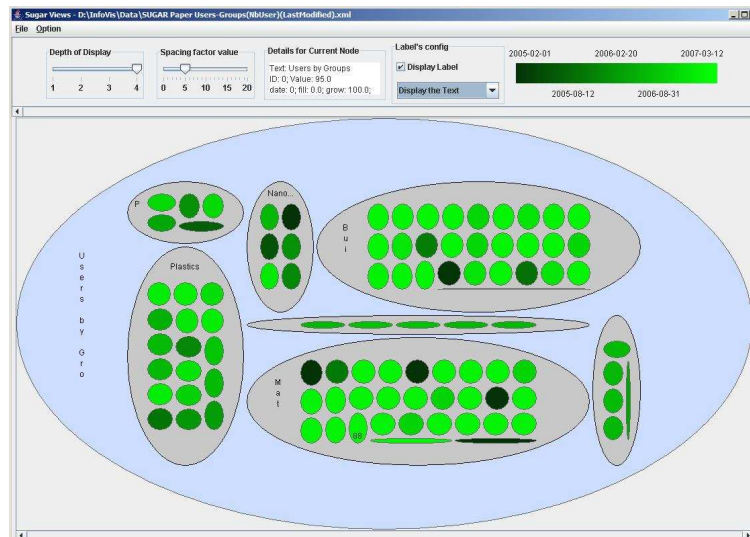


Fig. 179: Pseudo-coloriage selon une échelle mono-chromatique



Fig. 180: Pseudo-coloriage selon une échelle bi-chromatique

Une troisième option (cf. Fig. 181) permet d'appliquer une politique de coloriage dépendant d'un attribut des nœuds qui peut prendre valeurs discrètes (p.ex. type de média de communication, type de document). Dans cette configuration, une légende est affichée dans la partie supérieure de la fenêtre pour rappeler la signification de chaque couleur.

Il faut noter que cette fonctionnalité qui peut s'avérer très utile lorsque le nombre de valeurs discrètes possibles est limité (jusque 6 ou 8 valeurs), atteint ses limites pour des nombres plus élevés.

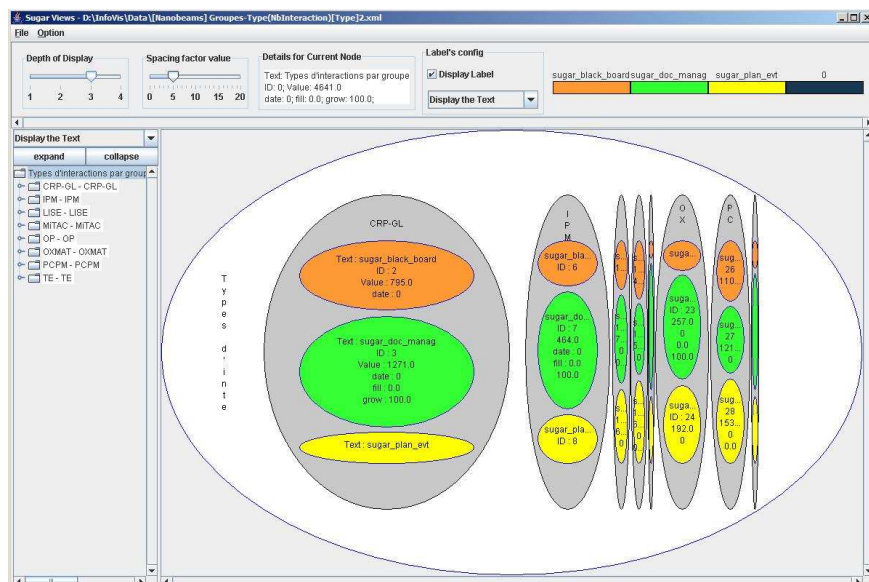


Fig. 181: Coloriage associé à une propriété à valeurs discrètes

La quatrième option est toujours associée à la visualisation des valeurs discrètes d'une propriété des nœuds mais au lieu de les associer à des couleurs distinctes, elle les représente par des images (cf. Fig. 182). Dans notre exemple, les valeurs 'low', 'medium' et 'high' sont respectivement associées aux images 'soleil', 'ciel nuageux' et 'ciel orageux'.

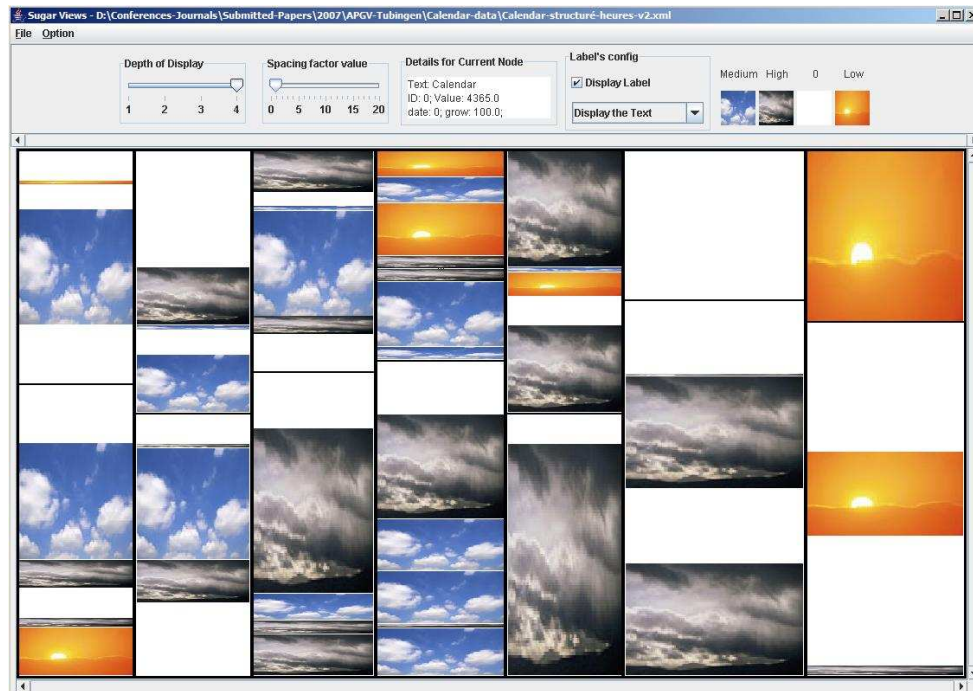


Fig. 182: Association d'images aux valeurs discrètes d'une propriété

Enfin la dernière possibilité permet de visualiser les feuilles de la hiérarchie par une icône particulière (cf. Fig. 183). Cette fonction se justifie lorsque la structure arborescente classe les éléments d'un ensemble. Dans ce cas, il peut s'avérer utile de visualiser différemment le dernier niveau hiérarchique (cf. discussion sur la loi de la taille relative dans la *Gestalt* théorie). Une telle approche permet en effet de différencier la structure de la classification et les éléments classifiés.

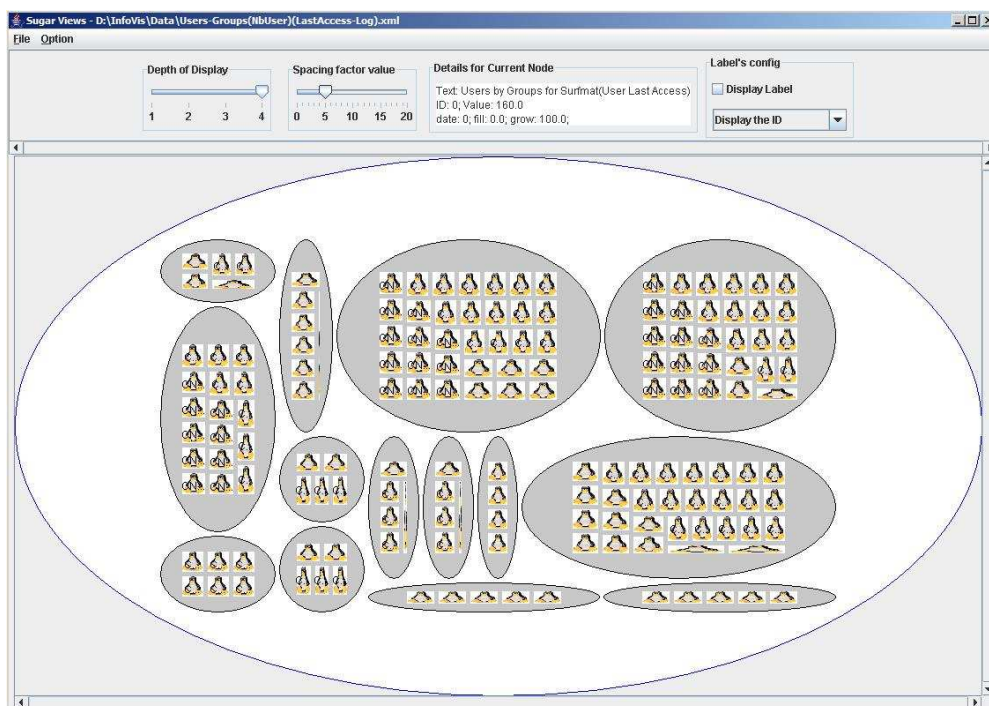


Fig. 183: Visualisation des feuilles de la hiérarchie par une icône

1.6 Ellimap-réservoirs

Le prototype offre différentes options pour configurer la représentation graphique des ellimaps-réservoirs (cf. p. 228).

La première concerne le mode de calcul du niveau de remplissage.

Nous avons décrit précédemment la procédure de calcul du niveau exact de remplissage par la méthode des splines cubiques (cf. p. 232). Cette approche est évidemment disponible dans le prototype. Cependant, nous avons également joint la possibilité de calculer le niveau de remplissage comme une fonction linéaire directement proportionnelle à la longueur de l’axe vertical de l’ellipse (comme il faudrait le faire dans un rectangle).

Le calcul par fonction linéaire induit naturellement une erreur d’un point de vue géométrique mais il est possible qu’elle améliore la perception du niveau de remplissage dans certaines circonstances. Cette question mériterait d’être approfondie dans de futurs travaux.

Le deuxième aspect qui a été particulièrement soigné dans le prototype concerne la politique de coloriage des ellimaps-réservoirs (cf. Fig. 184).

A ce niveau, nous distinguons deux concepts : le jeu de données hiérarchiques complet d’une part et la partie de celui-ci qui est affichée à l’écran sous la forme d’une ellimap d’autre part. Cette précision nous amène à distinguer cinq types de nœuds :

- les feuilles de la hiérarchie complète (*Tree Leaf*),
- les feuilles de la hiérarchie visualisée dans l’ellimap (*Visual Leaf*),
- les nœuds intermédiaires (*Tree nodes*),
- la racine de la hiérarchie complète (*Tree Root*),
- la racine de la hiérarchie visualisée dans l’ellimap (*Visual Root*).

Pour chacun des types de nœud, la couleur de la surface associée aux ressources consommées (cf. niveau de liquide) peut être choisie indépendamment des autres.

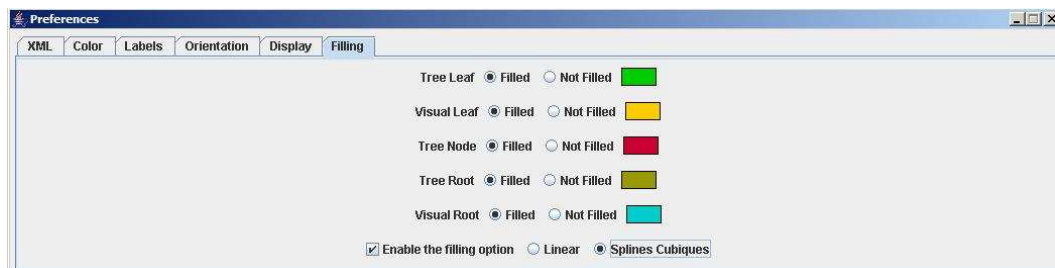


Fig. 184: Configuration des ellimaps réservoirs

Dans cette fonctionnalité, la variable visuelle ‘couleur’ n’est pas utilisée dans une stratégie de pseudo-coloriage mais pour différencier les nœuds selon leur position dans la hiérarchie. Par exemple, il est possible de savoir si un nœud qui apparaît comme une feuille au niveau visuel l’est réellement dans la hiérarchie complète.

Les Fig. 185 et Fig. 186 permettent de comprendre cette fonctionnalité. Elles correspondent à la configuration présentée dans la Fig. 184. Grâce aux couleurs⁵⁰, on remarque que la racine visuelle de la Fig. 185 est également la racine de la hiérarchie complète car elle est colorisée en ocre. Par contre, la racine visuelle de la Fig. 186 est un nœud intermédiaire de la hiérarchie complète car il est affiché en bleu clair. De manière similaire, on peut prendre conscience du

⁵⁰ Nous nous focalisons ici sur la politique de coloriage mais il est évident que si la vue en arbre est également affichée, il s’agit d’un autre moyen pour positionner un nœud de la vue ellimap dans la hiérarchie complète.

fait que certaines feuilles visuelles des deux figures sont des feuilles de la hiérarchie (couleur verte) et d'autres pas (couleur orange).

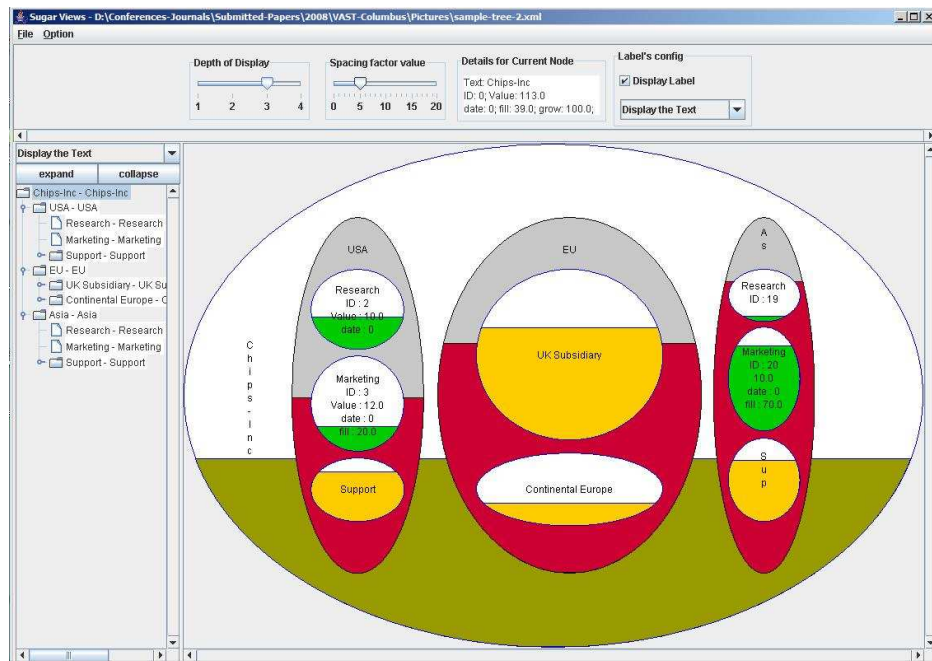


Fig. 185: Ellimap-réservoir: coloriage (à partir de la racine de la hiérarchie complète)

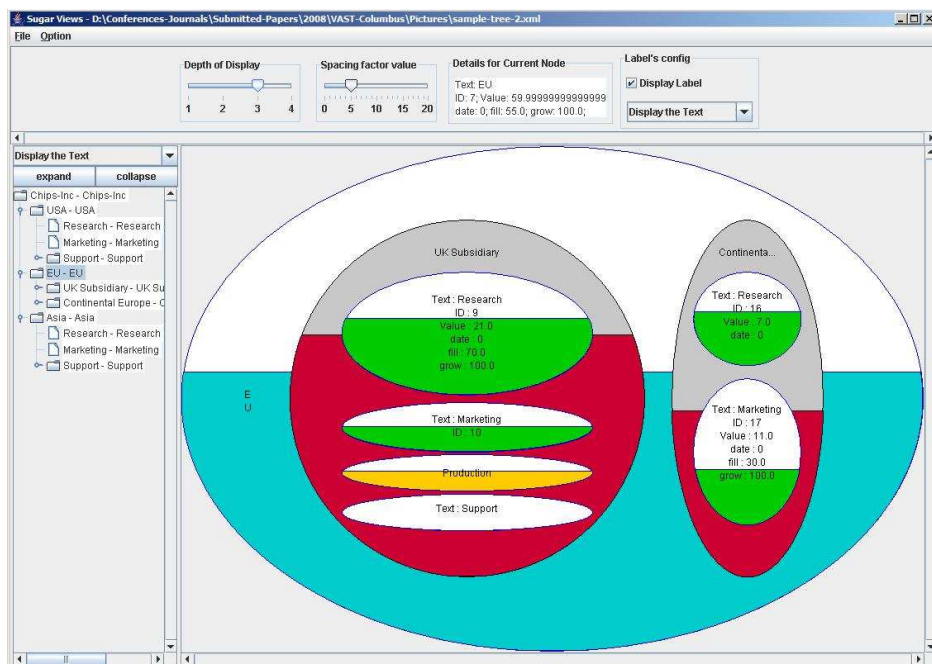


Fig. 186: Ellimap-réservoir: coloriage (à partir d'un nœud intermédiaire)

Outre l'aide au positionnement des nœuds affichés, il est évident que le fait de colorier la surface représentant les ressources consommées facilite également la perception de celle-ci. Il s'agit même de son premier avantage.

A titre de réflexion complémentaire, une évolution intéressante pourrait intégrer la technique du pseudo-coloriage dans les ellimaps-réservoirs.

1.7 Relation entre données et variables graphiques

Les données à visualiser sont fournies au prototype sous la forme d'un fichier XML. Ce format a été choisi car il implémente une structure hiérarchique et il est largement connu et disponible dans le monde informatique.

Le prototype est capable de visualiser tout document XML avec une vue par défaut. Cependant, certaines fonctionnalités nécessitent de préciser la relation entre certains attributs et les variables graphiques. En effet, alors que la représentation graphique de la seule structure du document XML n'exige pas de connaître les attributs des nœuds, la visualisation de leur poids demande de préciser quel attribut y fait référence.

Une fenêtre de configuration spécifique permet de définir les relations entre des fonctionnalités et des attributs du document XML. A titre d'exemples, dans la Fig. 187, le poids des nœuds est mentionné dans l'attribut 'value' et la propriété à valeur discrètes visualisable par une image par l'attribut 'image' (cf. Fig. 182).

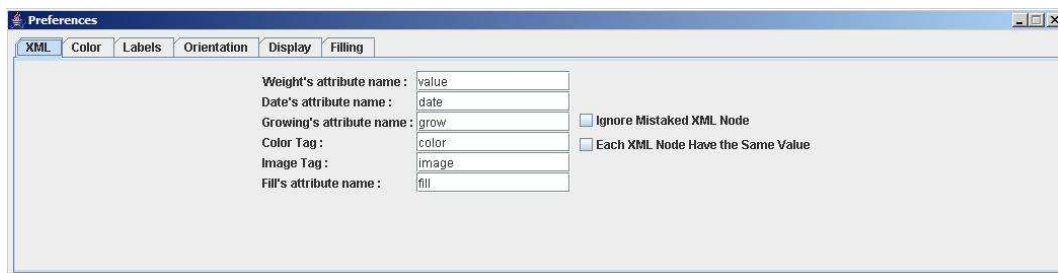
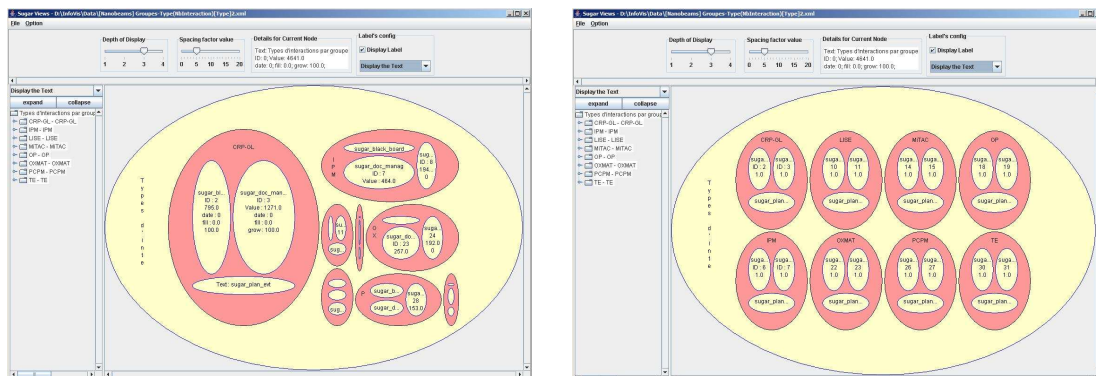


Fig. 187: Configuration de la relation entre le fichier de données XML et les variables graphiques

En complément de cette mise en relation, le prototype intègre une fonction permettant d'ignorer le poids des nœuds (cf. Fig. 188). Si elle est activée, cette option met en évidence la structure de la hiérarchie. Cela peut s'avérer utile pour les contextes où le poids des nœuds n'est pas la composante la plus importante à visualiser.



a. Prise en compte du poids des nœuds

b. Pas de prise en compte du poids des nœuds

Fig. 188: Fonction de désactivation de la prise en compte du poids des nœuds

Enfin, nous avons inclus la possibilité de ne pas afficher les nœuds pour lesquels un attribut à visualiser possède une valeur non conforme (p.ex. chaîne de caractères au lieu d'un nombre). Cette fonction s'avère inutile si le fichier XML a fait l'objet d'un passage par rapport à un schéma de référence mais ce n'est pas toujours le cas.

2. Utilisation pour visualiser la coopération

Le prototype peut être utilisé dans une multitude de contextes applicatifs. Compte tenu de l'objet de cette thèse, nous nous limiterons à illustrer son potentiel pour visualiser des informations associées à des plates-formes électroniques de coopération. Nous tenons néanmoins à insister sur son caractère générique.

Nous attirons également l'attention du lecteur sur trois points.

- Pour des raisons de confidentialité, les données associées à des plates-formes réelles de coopération électronique sont modifiées. Elles sont représentatives des informations présentes mais ne donnent pas d'information sur les données des instances réelles.
- Différentes version du prototype a été développées. Les copies d'écran sont issues de ces différentes instances.
- La gamme des possibilités du prototype pour aider à la supervision et à l'animation d'une plate-forme de coopération est large et nous n'avons inclus dans ce document qu'une sélection qui témoigne, selon nous, de ce potentiel.

Un premier exemple concerne la visualisation du nombre de clics par jour sur une certaine page d'un site web. Dans ce cas, une vue en *treemap* avec division alternée peut être utilisée (cf. Fig. 189). Elle conserve la séquence des nœuds et apparaît comme un calendrier déformé. Les colonnes sont associées aux mois de l'année visualisée et les rectangles de dernier niveau aux jours. Le poids des nœuds correspond au nombre de clics.

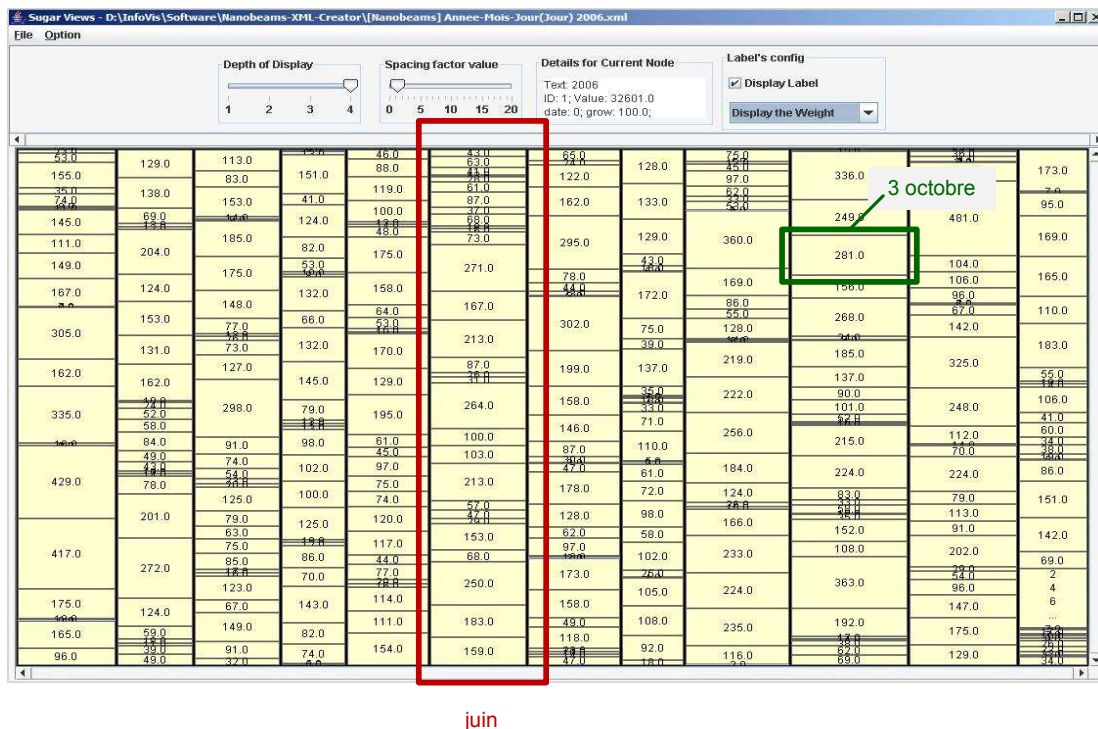


Fig. 189: Treemap indiquant le nombre de clics par jour sur une page web

Imaginons à présent que le prototype soit utilisé pour faire un suivi des activités de groupes d'utilisateurs sur une plate-forme coopérative. Nous pouvons être intéressés par représenter la taille relative des groupes d'utilisateurs ainsi que leur niveau d'activité. La comparaison des deux vues permet d'ailleurs de mieux comprendre ce qui se produit effectivement sur la plate-forme. Une possibilité de configuration du prototype utile dans ce contexte est fournie dans la

Fig. 190. Elle se base sur des ellimaps circularisées. Le poids des nœuds correspond respectivement au nombre d'utilisateurs dans chaque groupe et au nombre d'interactions avec la plate-forme générées par les membres de ce groupe. Nous pouvons observer en comparant les figures que la distribution des tailles des groupes diffère de celle de leur activité. Pour certains groupes l'activité est proportionnellement plus importante que leur taille et pour d'autres la relation inverse est observée.

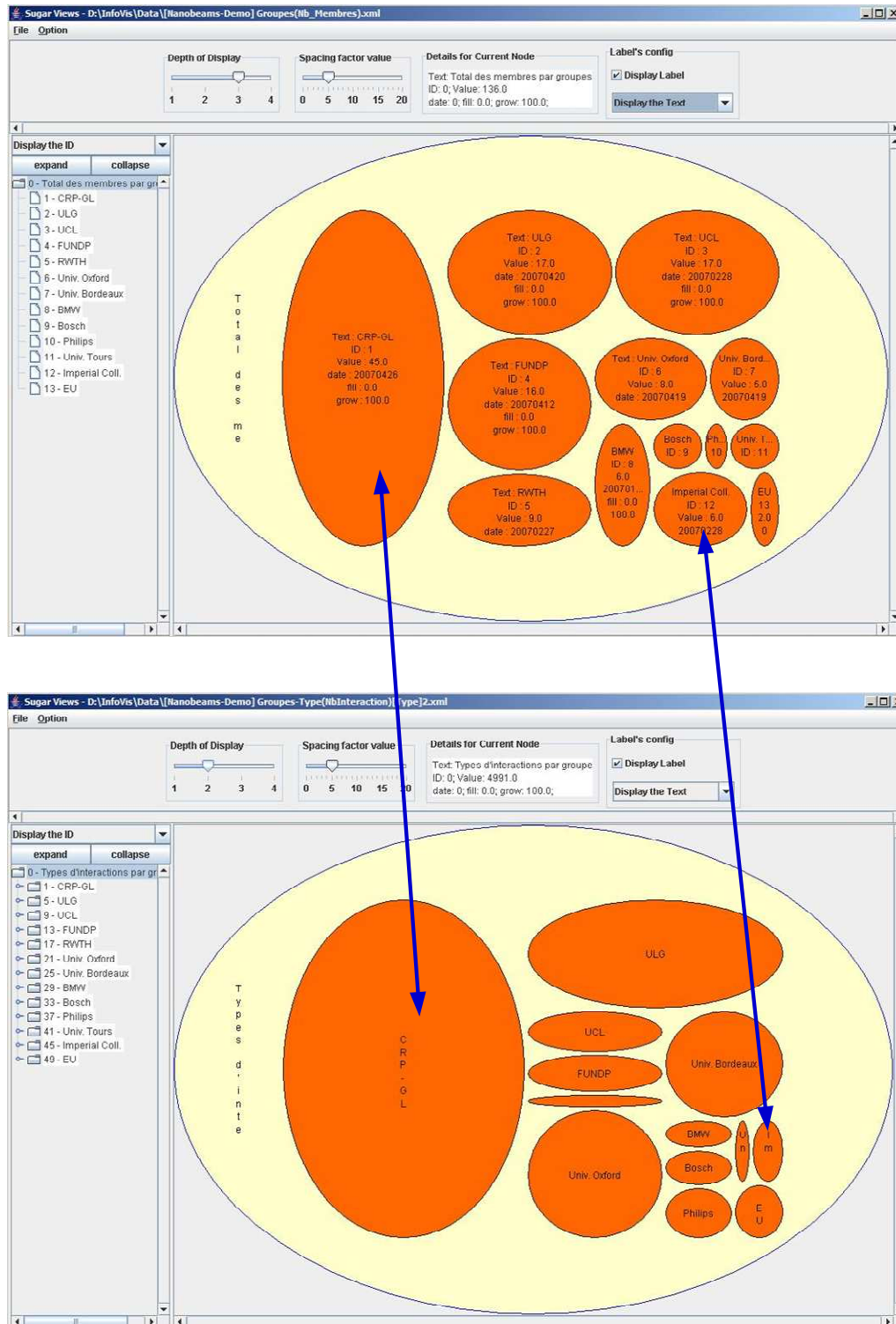


Fig. 190: Représentation de la taille et du volume d'activité des groupes

Le nombre d'interactions avec la plate-forme est naturellement un indice de leur niveau d'activité mais d'autres aspects peuvent aussi être importants à prendre en compte. Par exemple, la date de leur dernière interaction fournit également un élément d'interprétation utile. La Fig. 191 illustre ce cas particulier. Le poids des nœuds est associé au nombre d'interactions avec la plate-forme électronique de coopération. La date de la dernière interaction est mise en correspondance avec une échelle bi-chromatique 'rouge-vert'. Les interactions anciennes apparaissent donc dans le spectre des rouges et les nouvelles dans la gamme des verts.

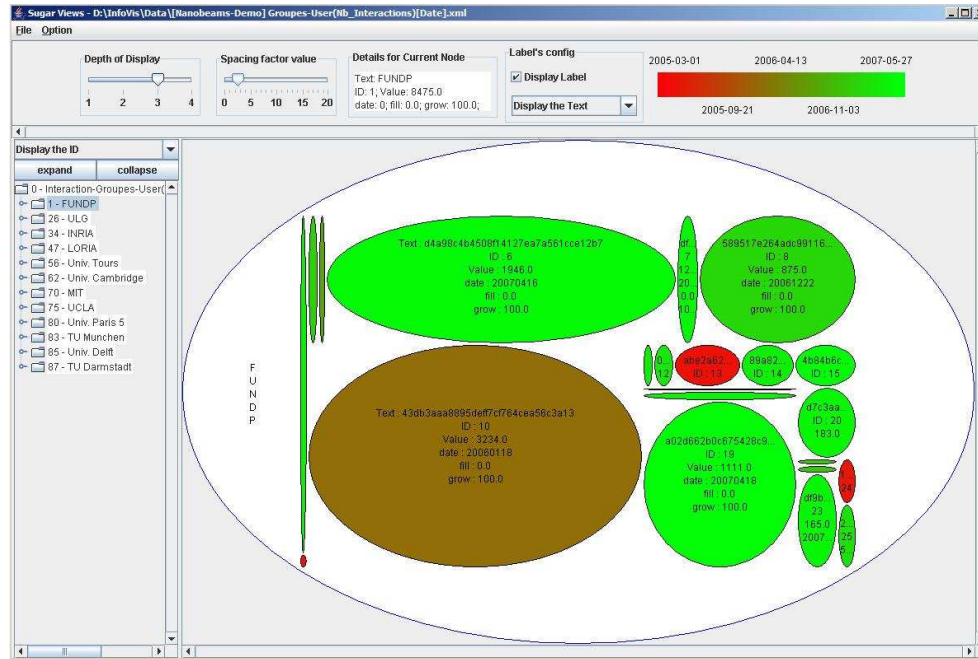


Fig. 191: Utilisateurs avec nombre d'interactions et date de la dernière d'entre elles

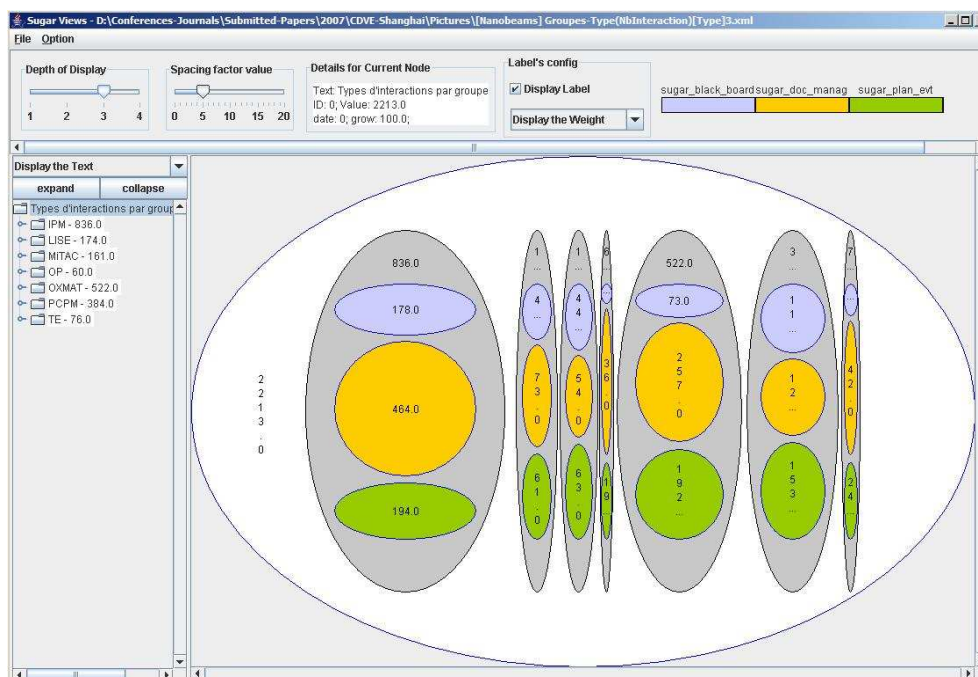


Fig. 192: Répartition de l'utilisation des outils coopératifs

La répartition de l'utilisation des différents outils présents sur la plate-forme électronique peut également s'avérer être un élément d'analyse pertinent des activités qui s'y déroulent. La visualisation de ce type d'information peut, par exemple, se baser sur la fonctionnalité de coloriage d'un attribut à valeurs discrètes. La Fig. 192 montre comment cette approche peut se matérialiser. Les différents outils de la plate-forme sont associés à une couleur (tableau noir → bleu, système de gestion de documents → orange, calendrier → vert).

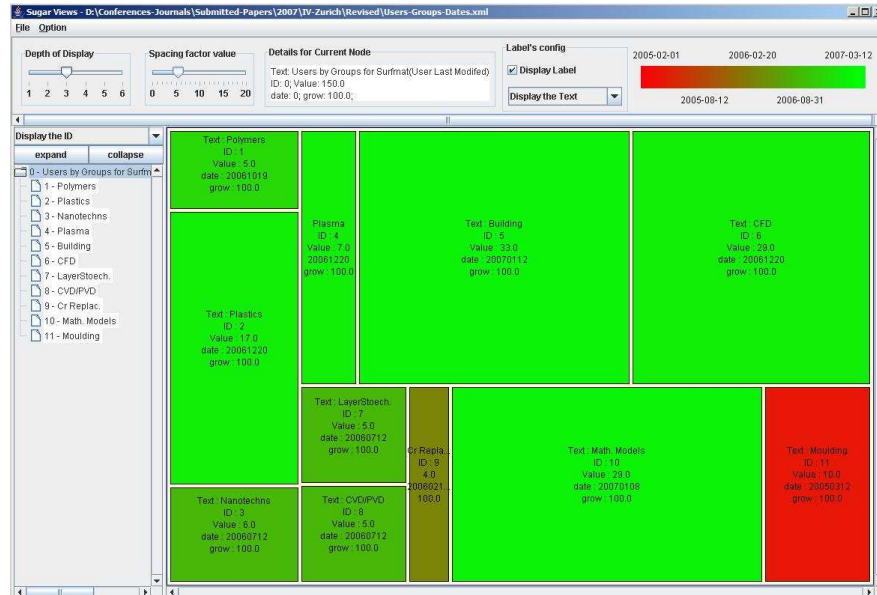


Fig. 193: Taille des espaces de travail et dernière activité enregistrée

Pour la personne qui doit superviser la répartition des quotas de stockage sur la plate-forme, une vue qui fournit pour chaque espace de travail sa taille et la date de la dernière interaction avec celui-ci est un outil d'analyse intéressant. Le prototype peut répondre à cette problématique en étant, par exemple, configuré comme à la Fig. 193. Une classique vue en *treemap* est utilisée dans laquelle la taille des espaces représente le poids des nœuds et la couleur est associée à la date de la dernière interaction.

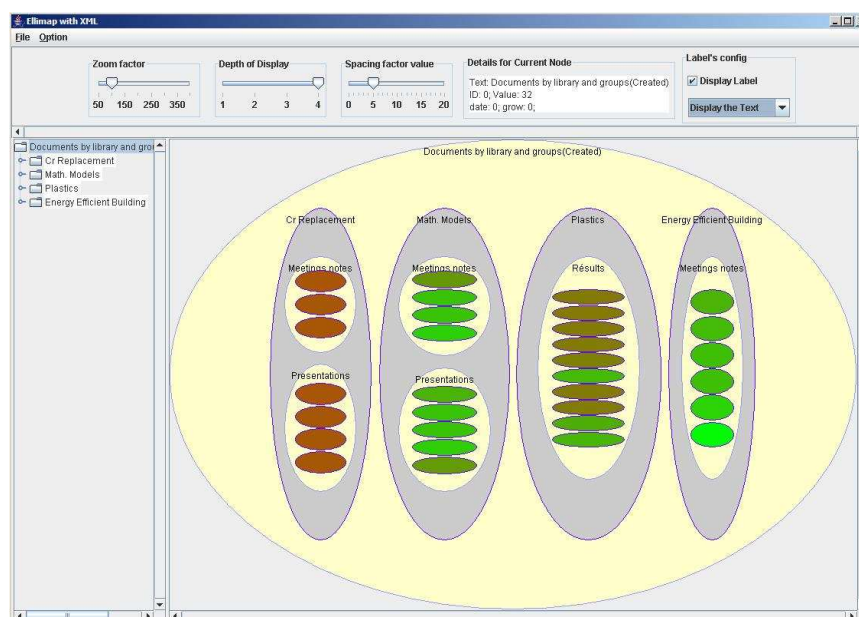


Fig. 194: Documents dans espaces de travail (avec dernière modification)

Les données à visualiser peuvent naturellement concerner des documents plutôt que des utilisateurs. Par exemple, on peut légitimement s'interroger sur le nombre de documents présents dans un ensemble d'espaces de travail et sur leur date de dernière modification. La Fig. 194 illustre une réponse possible à ce questionnement.

Dans certains cas, la dynamique des modifications de comportement est plus importante que leur valeur instantanée. Par exemple, si une plate-forme comporte plusieurs groupes d'utilisateurs, son superviseur peut vouloir connaître l'évolution du nombre de membres de chaque groupe. Nous présentons à la Fig. 195 une configuration du prototype dans laquelle la taille des groupes est associée au poids des nœuds et le taux de croissance de ceux-ci à une échelle bi-chromatique 'rouge-vert'. Si un groupe est en croissance, il apparaît en vert et s'il est en décroissance en rouge.

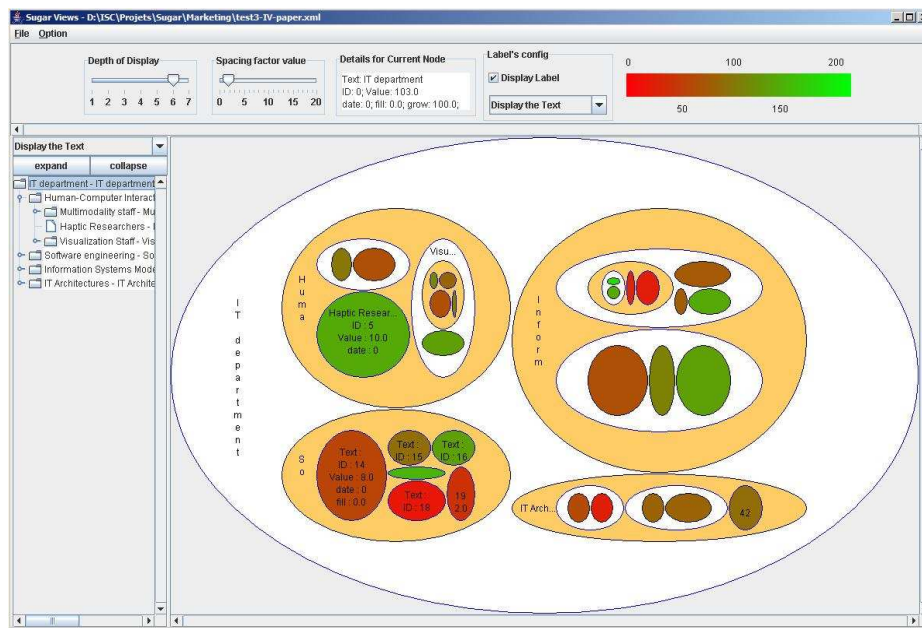
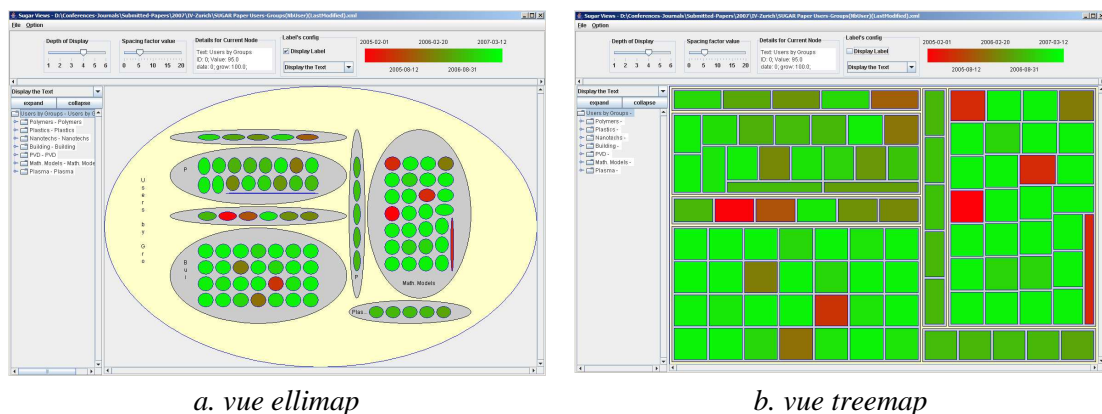


Fig. 195: Taille des groupes d'utilisateurs et évolution du nombre de leurs membres

En guise de conclusion, nous rappelons que selon la configuration du prototype utilisée, le résultat graphique peut être très différent (cf. Fig. 196). Il est de la responsabilité de la personne qui configure la vue de choisir le paramétrage qui correspond le mieux à ce qu'elle veut mettre en évidence.



a. vue ellimap

b. vue treemap

Fig. 196: Groupes d'utilisateurs et date de dernière connexion

Chapitre 10 : Conclusions et perspectives

« *Science sans conscience n'est que ruine de l'âme.* »

Rabelais

Selon notre perspective, les travaux réalisés au cours de cette thèse peuvent être regroupé en trois catégories : une étude approfondie de la problématique des taxonomies en visualisation d'information, une analyse de la question des métriques de l'activité coopérative et la conception et l'évaluation de la technique des ellimaps.

Au stade de la conclusion, il nous semble utile de rappeler les différents apports de nos travaux concernant chacun de ces points. Nous proposerons également des pistes qui, selon nous, mériteraient de faire l'objet de travaux complémentaires.

1. Taxonomies

Nos travaux ont débuté par une étude de l'état de l'art dans la discipline de la visualisation d'information. Très rapidement, nous avons perçu le besoin de disposer d'une taxonomie pour nous aider à structurer notre réflexion et à comparer les travaux de recherche antérieurs. En explorant cette question, nous avons pris conscience du fait que contrairement à d'autres disciplines, il n'existe actuellement pas de taxonomie de portée générale qui fasse l'objet d'un large consensus en matière de visualisation d'information. Nous avons recensé tout un ensemble de propositions dont certaines présentent d'indéniables qualités mais nous n'avons pas trouvé le Saint-Graal en ce domaine. Le simple fait que de multiples propositions coexistent et qu'aucune ne soit considérée comme référence témoigne du besoin d'unifier celles-ci dans un cadre conceptuel plus large. Nous aurions pu consacrer une thèse théorique complète à l'élaboration d'une telle taxonomie mais nous avons préféré axer nos recherches dans d'autres directions.

Nous avons d'abord estimé utile de comparer les différentes taxonomies recensées dans le domaine de la visualisation d'information. Ce travail nous a permis de proposer une méta-taxonomie qui identifie des propriétés génériques de ces taxonomies.

Ensuite, nous avons choisi de nous limiter à proposer une classification focalisée dont le domaine de définition se réduit aux techniques de visualisation d'activités coopératives.

A ce stade, nous avons perçu la nécessité de disposer d'un modèle de référence de ce que nous considérerons être une activité coopérative. Notre modèle extrêmement générique repose sur les concepts fondamentaux d'entités et d'interactions.

Après avoir approfondi la problématique des taxonomies d'un point de vue théorique, nous avons exploré la littérature relative à l'application de techniques de visualisation à des contextes coopératifs. Nous les avons regroupées en quatre grandes catégories selon la nature des objets visualisés : des communications, des actions sur des ressources partagées, la présence d'acteurs et des événements à caractère collectif. Nous avons aussi mis en évidence le besoin de considérer d'autres aspects, tels que le but de la visualisation, l'utilisateur cible ou le niveau d'intégration entre les outils de support à la coopération et de visualisation.

Nous disposions alors des objets conceptuels nous permettant de concevoir une taxonomie centrée sur la représentation graphique d'activités coopératives. Elle fut nommée « *IVF – Interaction Visualization Framework* ». Son élaboration, basée sur de multiples itérations, nous a obligé à la fois à réfléchir sur le concept même de taxonomie et à étudier de nombreuses techniques destinées à visualiser des informations relatives à des activités coopératives.

Grâce à ce travail conceptuel, nous nous sommes imprégnés de la discipline dans laquelle nous souhaitons entreprendre des recherches. Même si cela n'apparaît pas toujours de manière explicite dans les travaux ultérieurs, ces travaux initiaux constituèrent la base sur

laquelle repose nos propositions tant en matière de mesure de l'activité coopérative que de visualisation de celle-ci.

2. Activité coopérative

Nous avons choisi de situer nos travaux en visualisation d'information dans le contexte de la représentation d'activités coopératives.

Dans la section précédente, nous avons rappelé que notre première démarche a consisté à élaborer un modèle de la coopération basé sur l'idée d'entités en intractions.

Notre étude de la littérature nous a aidé à prendre conscience du rôle central joué par la notion de conscience de groupe dans les activités coopératives. Nous avons réfléchi à ce concept et avons proposé une catégorisation des informations qui y sont associées en fonction de deux dimensions : l'identification et la granularité. Les notions de conscience de groupe de l'espace de travail individuelle (WIA – *Workspace Individual Awareness*) et globale (WGA – *Workspace Global Awareness*) ont été déduites de cette analyse.

Afin de supporter la conscience de groupe par des moyens visuels, nous avons dû ensuite étudier les informations y relatives que nous pouvions représenter graphiquement. Ceci nous a amené à proposer des métriques du niveau de coopération informatisée au sein d'un groupe. Les indices *Qt*, *Qual*, *N-Qual* et *G* matérialisent le résultat de nos réflexions en ce domaine.

Très logiquement, nous avons ensuite cherché à visualiser ces indices dans le cas d'une plateforme électronique de coopération. Pour ce faire, nous avons successivement proposé deux approches. La première n'est pas innovante en termes de technique de visualisation. Elle consiste simplement à utiliser un graphique à deux axes orthogonaux pour représenter l'évolution de la valeur des indices au cours du temps. La seconde est plus originale. Elle repose sur la métaphore d'un drapeau pour visualiser, dans une zone d'attention périphérique, le niveau de conscience de groupe global (WGA).

Le suivi d'activités coopératives nécessite de visualiser non seulement un indice global mais aussi les interactions individuelles. Nous avons en partie répondu à ce besoin en imaginant la technique '*Clock View*'. Celle-ci se base sur la combinaison des métaphores de l'horloge et du groupe. Elle constitue une vue égo-centrique des interactions de l'utilisateur courant au cours d'une journée.

D'un point de vue général, ces techniques apportent un éclairage sur tout ou partie d'une situation de coopération électronique en facilitant la compréhension de paramètres déterminants d'un tel contexte.

3. Ellimaps

Au cours de notre étude des techniques de visualisation d'informations associées à des activités coopératives, nous avons été amenés à analyser le type de données à représenter. Nous avons alors pris conscience du fait que, dans un nombre significatif de cas, ces données se présentaient sous la forme d'une structure hiérarchique pondérée. Par conséquent, nous avons élargi notre réflexion à la problématique générale de la visualisation de ce type de données. Sur base de divers travaux antérieurs recensés dans la littérature, nous avons réalisé que les techniques de visualisation de ces structures de données se fondaient essentiellement sur deux propriétés visuelles : la clôture et la connectivité. Il nous est apparu que les premières, dont les *treemaps* constituent le plus illustre exemple, présentaient plus de potentiel que les secondes dans notre cas car elles supportaient la représentation d'un grand nombre de données et incluaient dans leur nature même la visualisation du poids des nœuds. Une revue approfondie des différentes variantes de *treemaps* a alors été entreprise. Sur cette base, nous avons identifié une lacune de cette classe de techniques : leur capacité limitée à faire percevoir la structure hiérarchique du jeu de données. Pour y répondre, diverses considérations, notamment basées sur des éléments de psychologie cognitive concernant la vision humaine, nous ont incités à imaginer la technique des ellimaps. Constatant certaines limitations de notre approche initiale, nous l'avons progressivement améliorée à divers points

de vue comme l'occupation de l'espace d'affichage ou la représentation de ressources consommables. Cette évolution progressive s'est fondée sur la combinaison de diverses évaluations formatives et de réflexions théoriques.

Dans un second temps, nous avons cherché à comparer de la manière la plus rigoureuse possible notre proposition (c.-à-d. l'état dans lequel se trouvait la technique des ellimaps à ce moment) avec une autre considérée dans la littérature comme un bon compromis (c.-à-d. les « *squarified treemaps* »). Ayant constaté le peu d'études comparatives similaires à celle que nous souhaitions entreprendre, nous avons passé de longues semaines à mettre au point une méthodologie que nous voulions la plus précise et argumentée possible. Par exemple, nous avons minutieusement élaboré une liste de tâches qui, selon nous, ne souffrait pas des biais observés dans d'autres études. De même, en essayant d'intégrer les recommandations formulées dans la littérature, nous en sommes arrivés à concevoir un logiciel dédié à nos tests.

Notre but était d'atteindre des résultats qui puissent être comparés à d'autres dans une démarche scientifique rigoureuse. En effet, en décrivant de manière très détaillée nos motivations, notre démarche intellectuelle, le choix des variables dépendantes et indépendantes, la procédure expérimentale, l'échantillon de sujets et enfin les résultats obtenus, nous fournissons à d'autres chercheurs les moyens d'infirmer ou de confirmer nos conclusions.

L'étude comparative s'est révélée conforme à notre intuition initiale. Le remplacement des rectangles par des ellipses dans une technique de visualisation basée sur la loi d'inclusion permet une meilleure perception de la structure arborescente des données sans pénaliser la perception du poids des nœuds.

4. Perspectives

Notre thèse, comme la plupart des travaux de recherche, pose presque autant de questions qu'elle n'apporte de réponses. Nous proposons à l'attention du lecteur quelques pistes ouvrant des perspectives dans lesquelles des travaux complémentaires pourraient être entrepris.

En premier lieu, nous pensons que des travaux de nature plus théorique sont encore indispensables sur le thème des taxonomies des techniques de visualisation d'information. A nos yeux, les chercheurs du domaine manquent encore d'un cadre conceptuel général et universellement accepté pour positionner leurs travaux les uns par rapport aux autres. Nous ne pouvons affirmer à ce stade de notre réflexion si une classification aussi générique que la tableau de Mendeleev pourra un jour être élaboré, si nous verrons plutôt apparaître une classification globalement stable bien que régulièrement remise à jour telle que celles utilisées en systématique botanique ou si la nature même des objets à classer empêchera l'émergence d'une taxonomie générale. Seuls des travaux complémentaires pourront fournir un éclairage sur ce point.

Deuxièmement, notre étude bibliographique sur le travail coopératif assisté par ordinateur nous a permis de prendre conscience du rôle central joué par le concept de conscience de groupe. Un consensus émerge cependant dans la littérature pour admettre que des travaux de recherche sont encore nécessaires pour mieux comprendre quelle information doit être communiquée, à quel moment et sous quelle forme pour supporter efficacement l'émergence et le maintien d'une conscience de groupe par des moyens informatiques.

Un type particulier de conscience de groupe, celle associée à un espace de travail (*workspace awareness*), mérite d'être complété au niveau théorique pour évaluer dans quelle mesure les concepts élaborés dans les travaux relatifs à la conscience de groupe d'un espace de travail peuvent être étendus à un environnement mixte combinant différents espaces de coopération à vocation professionnelle et des espaces de communication et d'échange appartenant à la sphère privée. L'étude de ce contexte plus complexe nous semble importante car, dans certaines circonstances, le comportement de l'utilisateur au sein de l'espace de travail est directement ou indirectement influencé par l'état de sa situation dans d'autres communautés appartenant notamment à la sphère privée. La Fig. 197 illustre une situation relativement habituelle dans laquelle une personne doit rester informée au niveau professionnel des

événements associés à des plates-formes virtuelles de coopération (A, B), à des projets qui ne sont pas supportés par de telles plates-formes (K, L) et à une équipe focalisée sur un thème particulier en dehors du cadre d'un projet (T). Pour certaines décisions, telles que la fixation d'une réunion, la planification d'un déplacement ou l'imposition d'un délai, cette personne sera influencée par des événements appartenant à sa sphère privée. Dans l'exemple, il pourra s'agir d'événements liés à sa famille (H), à des amis (F) ou à des activités de loisirs (M, N). La notion de conscience de groupe relative à l'environnement de travail doit donc être complétée par un concept plus large de conscience de l'environnement global de l'utilisateur.

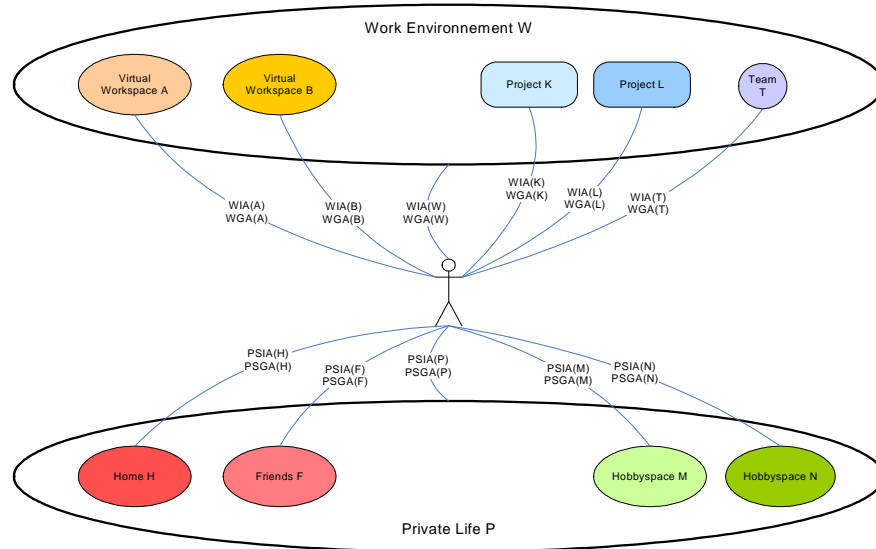


Fig. 197: Contexte hybride : professionnel et privé

Naturellement, la totalité des informations associées à tous ces environnements ne peut être assimilée rapidement par l'utilisateur. Des questions importantes se posent donc en termes de choix et de granularité des informations à communiquer. Des questions théoriques se posent également en termes de combinaison d'informations appartenant à des contextes sociaux et psychologiques différents.

Troisièmement, bien que la technique des ellimaps ait démontré un potentiel certain, nous sommes convaincus qu'il est possible de l'améliorer encore à divers points de vue. L'occupation de l'espace d'affichage pourrait certainement être encore optimisée. En fait, en théorie, chaque algorithme d'amélioration des *treemaps* pourrait être adapté au cas des ellimaps pour en déterminer l'applicabilité à ce contexte. De même, une approche exhaustive demanderait de réaliser pour chaque variante des ellimaps une évaluation comparative avec chaque variante des *treemaps*. Nous sommes néanmoins conscients de l'importance des ressources qu'une telle étude exigerait.

Dans une optique plus appliquée, il est évident que la technique des ellimaps devrait être confrontée à des données issues de multiples contextes applicatifs afin d'en identifier la portée complète. Est-elle adaptée à la représentation de données dans le secteur du marketing, des sciences de la vie ou de l'ingénierie ? Nous n'avons pu l'évaluer en pratique. Il s'agit là cependant d'une des pistes que nous continuerons d'explorer à l'avenir.

En guise de conclusion, il est évident que nous n'avons pas pu aborder dans cette thèse toutes les questions que nous nous sommes posées au cours des dernières années. Plusieurs fois, nous avons dû abandonner des idées qui nous semblaient prometteuses afin de nous recentrer sur des thématiques précises. C'est là le lot de toute entreprise de recherche et nous nous y sommes donc soumis, parfois avec regret mais en restant toujours conscients que cette rigueur dans la gestion du temps était indispensable pour arriver à bon port dans le délai imparti.

5. Postface

« *Le commencement de toutes les sciences,
c'est l'étonnement de ce que les choses sont ce qu'elles sont.* »

Aristote

Une thèse, outre sa contribution à la production de connaissances nouvelles, doit à nos yeux également s'inscrire dans une démarche d'ouverture d'esprit, d'originalité dans les idées et d'humanisme au sens le plus large. Avant de le quitter, nous livrons donc au lecteur quelques idées plus déroutantes. Nous ne les inscrivons pas dans une démarche scientifique rigoureuse mais nous les apportons sur l'agora en vue de susciter des conversations que nous souhaitons animées, contradictoires et constructives.



Fig. 198: Jardin Ninomaru, Kyoto, Japon

A la fin de notre thèse, une idée inédite, déroutante et que certains jugeront probablement sans intérêt, a émergé dans notre esprit. Nous avons pris conscience du parallélisme surprenant entre nos travaux sur la visualisation de données hiérarchiques et certains éléments de la théorie du paysage. A titre privé, notre passion pour la conception de jardins nous amène régulièrement à lire divers ouvrages tant sur la théorie de l'élaboration d'un jardin que sur les différentes plantes que l'on peut y placer. Dans ce contexte, « *L'optique des jardins* » de Robert Mallet (2004) est un des ouvrages que nous considérons comme les plus didactiques sur la question. L'auteur y explique comment utiliser les éléments visuels tels que la structure des végétaux, les ombres, les couleurs, les courbes et les droites ou encore la perspective pour concevoir un jardin harmonieux.

C'est évidemment le même œil qui admire un jardin ou tente d'extraire une sémantique d'un graphique de données. Ses capacités physiques, les processus cognitifs élémentaires auxquelles il fait appel sont semblables. Bien sûr, les circonstances ne sont pas comparables et certains paramètres de cette chaîne complexe de traitement de l'information diffèrent dans les deux cas.

Cependant, nous livrons à l'esprit du lecteur curieux certaines observations que nous estimons dignes d'une réflexion critique.

Tout d'abord, nous avons justifié dans cette thèse pourquoi l'inclusion successive de surfaces courbes offrait un potentiel plus important que celle de surfaces à angles vifs d'un point de vue perceptif. Nous laissons le lecteur juger de l'intérêt de mettre en relation cette

argumentation avec l'extrait suivant de l'ouvrage de Mallet : « *On entend parfois dire que la ligne courbe prolonge le regard, alors que la ligne droite l'enferme. Propos subjectifs pour certains ? Plutôt subversifs pour d'autres ! La courbe nous invite avec douceur à suivre son cheminement. La droite par sa raideur et sa dureté repousse le regard, elle est dans un sens menaçante, synonyme de danger, d'interdit.* » (p. 10)

Ensuite, la conception des ellimaps prit naissance dans une intuition selon laquelle les variables graphiques à considérer en priorité étaient celles conditionnant le positionnement dans l'espace d'affichage. Les autres ne représentent que l'habillage d'une structure graphique préalablement définie. La couleur, par exemple, par sa puissance visuelle impressionnante, peut complètement biaiser la perception d'un positionnement inadéquat. Dès lors, si elle est capable de compenser les lacunes d'une disposition spatiale sub-optimale, quelle ne doit pas être son rayonnement dans une disposition adéquate ! De même en est-il dans le secteur du paysage. « *Il en va de même d'un paysage encombré, lorsqu'on y perd son latin. Pour y voir plus clair, il suffit alors de se promener au clair de lune. Apparaissent alors les formes et les structures prépondérantes. On éliminera alors les détails superflus par un bon usage de la tronçonneuse. Pour obtenir le même effacement des signes inutiles les peintres du XVII^{ème} siècle utilisaient une technique qui consistait à regarder à travers un verre fumé pour ne retenir que les sujets dignes d'être peints.* » (p. 18).

Enfin, un grand nombre de chercheurs en visualisation ont pour objectif de maximiser le nombre d'éléments d'information affichables dans un espace limité. Nous ne remettons pas en cause la pertinence de ce défi. Nous invitons simplement le lecteur à aborder la question sous un autre point de vue. Nous avons expliqué dans cette thèse, quand nous avons décrit les ellimaps-réservoirs, que l'espace libre entre les formes géométriques incluses les unes dans les autres était digne d'intérêt en tant que tel. A nos yeux, l'espace inoccupé, le fond de la représentation graphique ne peut être considéré comme quantité négligeable qu'il importe de réduire à sa plus simple expression. Au lecteur à l'esprit ouvert, nous proposons donc de méditer sur la correspondance de cette idée avec l'extrait suivant : « *Dans un orchestre, les cordes sont placées traditionnellement devant les instruments à vent. En gastronomie, les fonds de cuisine sont le secret des bonnes sauces. L'art de recevoir commande de choisir une compagnie agréable pour mettre en valeur les invités de marque. Il en est de même pour les jardins : les fonds doivent faire l'objet d'un soin très attentif. Ils ne doivent être ni uniformes ni d'une même teinte.* » (p. 85)

*Que ces idées soient perçues provocatrices, nous en serions flattés.
Qu'elles soient considérées subversives, nous nous en étonnerions.
Qu'elles suscitent d'âpres débats, n'est pas là la nourriture de l'esprit ?
Qu'on les réfute, enfin la science les rejoindrait.
Qu'elles soient ignorées, voilà notre seule crainte.*

Benoît Otjacques, Launoy, 19 juillet 2008

Chapitre 11 : Bibliographie

- [1] Adams M., Tenney Y. et Pew R. (1995). *Situation Awareness and the Cognitive Management of Complex Systems*. Human Factors, Vol. 37, N° 1, pp. 85-104.
- [2] Association Française de Normalisation (AFNOR) (1995). *Etude de la normalité d'une distribution*, norme NF X06-050.
- [3] Andrews K. (2006). *Evaluating Information Visualisations*. Proceedings of the 'Beyond time and errors: novel evaluation methods for Information Visualization' workshop (BELIV 2006), dans le cadre de la conférence AVI 2006, 23 mai 2006, Venise, Italie.
- [4] Andrews K. (2007). *A Comparative Study of Four Hierarchy Browsers using the Hierarchical Visualisation Testing Environment (HVTE)*. Proceedings of the 11th International Conference Information Visualization (IV'08), 4-6 Juillet 2007, Zurich, Suisse, pp. 81-86.
- [5] Andrews K. (2007). *Usability Evaluation & Information Visualization*. Tutorial, 11th International Conference Information Visualization (IV'08), 4-6 Juillet 2007, Zurich, Suisse.
- [6] Apperley M., Fletcher D., Rogers B. et Thomson K. (2000). *Interactive Visualisation of a Travel Itinerary*. Proceedings of the working conference on Advanced Visual Interfaces (AVI'00), 23-26 Mai 2000, Palerme, Italie, pp. 221-226.
- [7] Ball R., North C. et Bowman D.A. (2007). *Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'07), 28 Avril-3 Mai 2007, San Jose, California, USA, pp. 191-200.
- [8] Balzer M, Deussen O. et Lewerentz C. (2005). *Voronoi Treemaps for the Visualization of Software Metrics*. Proceedings of the ACM Symposium on Software Visualization 2005, 14-15 Mai 2005, St. Louis, USA, pp. 165-172.
- [9] Bannon L. et Schmidt K. (1991). *CSCW: four characters in search of a context*. Studies in CSCW. Theory, practice and design. Bowers J.M. Benford S.D., North Holland, pp. 3-15.
- [10] Bannon L.J. et Karsenty L. (1997). *From cognitive ergonomics to CSCW: extension or reformulation ?* Working Paper, COTCOS project, accessible à l'adresse Internet: <http://www-sv.cict.fr/cotcos/pjs/>, accédé le 23 septembre 2005.
- [11] Barros B. et Verdejo M.F. (1999). *An Approach to analyse collaboration when shared structured workspaces are used for carrying out group learning processes*. Proceedings of the Artificial Intelligence In Education Conference (AIED'99), 19-23 Juillet 1999, Le Mans, France.
- [12] Baudel T. (1995). *Aspects morphologiques de l'interaction humain – ordinateur: Etude de modèles d'interaction gestuels*. Thèse de doctorat soutenue le 15 décembre 1995 à l'Université de Paris Sud.
- [13] Baudel T. (2002a). *Visualisations Compactes: Une Approche Déclarative pour la Visualisation d'Information*. Actes de la 14^{ème} Conférence Francophone sur l'Interaction Homme-Machine (IHM 2002), (14th French-speaking Conference on HCI), 26-29 Novembre Septembre 2002, Poitiers, France.
- [14] Baudel T. (2002b). *Canonical Representation of Data-Linear Visualization Algorithms and its Applications*. ILOG Technical Report Nr 02-2002.
- [15] Beale M., Einstein M., McCrickard S., North C. et Saraiya P. (2001). *Visualizing Communication Timelines Containing Sparsely Distributed Clusters*. Proceedings of IEEE Symposium on Information Visualization, 22-23 Octobre 2001, San Diego, Californie.
- [16] Bederson B., Clamage A., Czerwinski M. et Rebertson G. (2003). *A Fisheye Calendar Interface for PDAs: Provinding Overviews for Small Displays*. Proceedings of the

Conference on Human Factors in Computing Systems (CHI'03), 5-10 Avril 2003, Ft. Lauderdale, Floride, USA, pp. 618-619.

[17] Bederson B., Clamage A., Czerwinski M. et Robertson G. (2004). *Date Lens: A Fisheye Calendar Interface for PDAs*. ACM Transactions on Computer-Human Interaction, Vol. 11, N° 1, Mars 2004, pp. 90-119.

[18] Begole J., Tang J. C., Smith R. B. et Yankelovich N. (2002). *Work Rhythms: Analysing Visualizations of Awareness Histories of Distributed Groups*. Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'02), 16-20 Novembre 2002, New Orleans, Louisiane, USA, pp. 334-343.

[19] Begole J., Tang J., et Hill R. *Rhythm Modeling, Visualizations and Applications*. Proceedings of the 16th annual ACM symposium on User Interface Software and Technology (UIST'03), 2-5 November 2003, Vancouver, Canada, pp. 11-20.

[20] Bellotti V., Ducheneaut N., Howard M. et Smith I. (2003). *Taking Email to Task: The design and Evaluation of a Task Management Centered Email Tool*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'03), 5-10 Avril 2003, Ft. Lauderdale, Floride, USA, pp. 345-352.

[21] Benford S., Bowers J., Fahlen L., Greenhalgh C. et Snowden D. (1995). *User Embodiment in Collaborative Virtual Environments*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'95), 7-11 Mai 1995, Denver, Colorado, USA, pp. 242-249.

[22] Bertin J. (1967), *La sémiologie graphique: les diagrammes, les réseaux et les cartes*, Editions Gauthier Villars, Paris, 1967, p.8.

[23] Bertin J. (1998). *Sémiologie graphique*, réimpression de 1998 de la 2^{ème} édition de 1973, Editions des hautes Etudes en Sciences Sociales, Paris.

[24] Bertin J. (2000). *Brève présentation de la graphique*, cours disponible à l'adresse Internet : <http://www.sciences-po.fr/cartographie/>, accédé le 12 janvier 2005.

[25] Blanch R. et Lecolinet E. (2007). *Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques*. IEEE Transactions on Visualization and Computer Graphics, Vol. 13, N°6, Novembre/Décembre 2007, pp. 1248-1253.

[26] Boardman R. (2000). *Bubble Trees. The Visualization of Hierarchical Information Structures*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'2000), 1-6 Avril 2000, La Haye, Pays-Bas, pp. 315-316.

[27] Borges M.R.S. et Pino J.A. *Awareness Mechanisms for Coordination in Asynchronous CSCW*. Proceedings of the 9th Workshop on Information Technologies and Systems (WITS'99), 11-12 Décembre 1999, Charlotte, Caroline du Nord, USA, pp. 69-74.

[28] Boyd D. (2002), *Faceted Id/Entity: Managing representation in a digital world*, Master thesis, Massachusetts Institute of Technology, présentée en septembre 2002.

[29] Boyd D. et Potter J. (2003), *Social network fragments: an interactive tool for exploring digital social connections*. Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2003), 27-31 Juillet 2003, San Diego, Californie.

[30] Brézillon P., Borges M.R.S., Pino J.A. et Pomerol J.-C. (2004). *Context-Awareness in Group Work: Three Case Studies*. Proceedings of the IFIP International Conference on Decision Support Systems (DSS 2004), 1-3 Juillet 2004, Prato, Italie, pp. 115-124.

[31] Bruley C. (1999), *Analyse des représentations graphiques de l'information – extension aux représentations tridimensionnelles*. Thèse de doctorat en informatique, soutenue en juin 1999, Université Joseph Fourier de Toulouse.

- [32] Bruls D. M., Huizing C. et van Wijk J. J. (2000). *Squarified Treemaps*. Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization 2000, 29-31 Mai 2000, Amsterdam, Pays-Bas, pp. 33-42.
- [33] Butler D. M. et Hansen C. (1992). *Visualization'91 Workshop Report: Scientific Visualization Environments*. Computer Graphics, Vol. 26, N° 3, Août 1992, pp. 213-216.
- [34] Cadiz J.J., Venolia G., Jancke G. et Gupta A. (2002). *Designing and Deploying an Information Awareness Interface*. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'02), 16-20 Novembre 2002, New Orleans, Louisiane, USA, pp. 314-323.
- [35] Card S. K. et Mackinlay J. (1997). *The Structure of the Information Visualization Design Space*. Proceedings of the IEEE Symposium on Visualization (InfoVis' 97), 18-25 Octobre 1997, Phoenix, Arizona, USA, pp. 92-100.
- [36] Card S. K., Mackinlay J. D. et Shneiderman B. (1999). *Readings in Information Visualization, Using Vision to Think*. Morgan Kaufmann Publishers, Academic Press, San Diego, pp. 6-7.
- [37] Card S. (2003). *Information Visualization*. dans J.A. Jacko, A. Sears (Eds): *The Human-Computer Interaction Handbook, Fundamentals, Evolving Technologies and Emerging Applications*, Lawrence Erlbaum Associates Publishers, London, p. 551.
- [38] Chasseur H. (2002), *Visualisation interactive par un graphe hypermédia des relations entre acteurs, activités et documents au cours de la conception d'un bâtiment*. Mémoire de DEA CRAI, Ecole d'architecture de Nancy, France.
- [39] Chi E. H. (2000). *A Taxonomy of Visualization Techniques using the Data State Reference Model*. Proceedings of IEEE Symposium on Visualization (InfoVis' 2000), 9-10 Octobre 2000, Salt Lake City, Utah, USA, pp. 69-75.
- [40] Chi E. H., Pitkow J., Mackinlay J., Pirolli P., Gossweiler R. et Card S. K. (1998). *Visualizing the Evolution of Web Ecologies*. Proceedings of the Conference on Human Factors in Computing Systems 1998 (CHI'98), 18-23 Avril 1998, Los Angeles, USA, pp. 400-407.
- [41] Cockton G., Lavery D. et Woolrych A. (2003). *Inspection-based evaluations*. Jacko J.A., Sears A. (Eds): *The Human-Computer Interaction Handbook, Fundamentals, Evolving Technologies and Emerging Applications*, Lawrence Erlbaum Associates Publishers, London, p. 1118-1138.
- [42] Dachsel R. et Weiland M. (2006). *TimeZoom: A Flexible Detail and Context Timeline*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'06), 22-27 April 2006, Montréal, Canada, pp. 682-687.
- [43] Daft R. L. et Lengel R. H. (1984). *Information richness: a new approach to managerial behavior and organizational design*. Cummings, L. L. et Staw, B. M. (eds), *Research in organizational behavior*, Vol. 6, Homewood, JAI Press, pp. 191-233.
- [44] Daft R. L. et Lengel R. H. (1986). *Organizational information requirements, media richness and structural design*. *Management Science*, Vol. 32, N° 5, pp. 554-571.
- [45] Dagnelie P. (1998) *Statistique théorique et appliquée, tome 1, Statistique descriptive et bases de l'inférence statistique*. Editions De Boeck Université, Bruxelles.
- [46] Dagnelie P. (2006). *Statistique théorique et appliquée, vol. 2 Inférence statistique à une et à deux dimensions* (2^{ème} Edition). Editions De Boeck Université, Bruxelles.
- [47] Decortis F., Noirfalise S. et Saudelli B. (1997). *Distributed Cognition as Framework for Cooperative Work*. Working Paper, COTCOS project, accessible à l'adresse Internet: <http://www-sv.cict.fr/cotcos/pjs/>, accédé le 15 septembre 2005.

- [48] Decortis F., Noirfalise S. et Saudelli B. (1997). *Cognitive Ergonomics as Framework for Cooperative Work*. Working Paper, COTCOS project, accessible à l'adresse Internet: <http://www-sv.cict.fr/cotcos/pjs/>, accédé le 23 septembre 2005.
- [49] Demian P. et Fruchter R. (2006a) *Finding and understanding reusable designs from large hierarchical repositories*. Information Visualization, 2006, Vol. 5, pp. 28-46.
- [50] Demian P. et Fruchter R. (2006b) *A Methodology for Usability Evaluation of Corporate Memory Design Reuse Systems*. ASCE Journal of Computing in Civil Engineering, Vol. 20, N° 6, pp. 377-389.
- [51] de Terssac G. et Chabaud C. (1991). *Référentiel commun et fiabilité*. Leplat J. et de Terssac G. (Eds) : Les facteurs humains de la fiabilité dans les systèmes complexes, Toulouse, Editions Octarès, pp. 111-140.
- [52] Dix A., Beale R. et Wood A. (2000). *Architectures to make Simple Visualisations using Simple Systems*. Proceedings of the ACM working conference on Advanced Visual Interfaces 2000 (AVI 2000), 23-26 Mai 2000, Palerme, Italie, pp. 51-60.
- [53] Donath J. (2002). *A semantic Approach to Visualizing Online Conversations*. Communications of the ACM, Vol. 45, N° 4, Avril 2002, pp. 45-49.
- [54] Donath J., Karahalios K. et Viégas F. (1999). *Visualizing Conversations*. Proceedings of 32th Hawaii International Conference on System Sciences (HICSS-32), 5-8 Janvier 1999, Island of Maui, Hawaii, USA.
- [55] Dourish P. et Bellotti V. (1992). *Awareness and Coordination in Shared Workspaces*. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'92), 1-4 November 1992, Toronto, Canada.
- [56] Dragicevic P. et Huot S. (2002). *SpiraClock: A continuous and Non-Intrusive Display for Upcoming Events*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'02), 20-25 Avril 2002, Minneapolis, Minnesota, USA, pp. 604-605.
- [57] Ducheneaut N. et Bellotti V. (2001). *E-mail as Habitat*. Interactions, Vol. 8 (5), pp. 30-38.
- [58] Dumas J. S. (2003). *User-based evaluations*. Jacko J.A., Sears A. (Eds): The Human-Computer Interaction Handbook, Fundamentals, Evolving Technologies and Emerging Applications, Lawrence Erlbaum Associates Publishers, London, p. 1093-1117.
- [59] Ellis C. (1994). *Keepers, Synchronizers, Communicators and Agents*. ACM SIGOIS Bulletin, Avril 1995, Vol. 15, N° 3, en référence au Workshop 'Software Architectures for Cooperative Systems', Conférence CSCW 94, 22-26 Octobre 1994, Chapel Hill, USA.
- [60] Ellis C., Gibbs S.J. et Rein G.L. (1991). *Groupware: Some Issues and Experiences*. Communications of the ACM, Vol. 43, N° 1, Janvier 1991, pp. 39-58.
- [61] Endsley M. (1995). *Toward a Theory of Situation Awareness in Dynamic Systems*. Human Factors, Vol. 37, N° 1, pp. 32-64.
- [62] Erickson T., Halverson C., Kellogg W. A., Laff M. et Wolf T. (2002). *Social translucence, designing social infrastructures that make collective activity visible*. Communications of the ACM, Avril 2002, Vol. 45, N° 4, pp. 40-44.
- [63] Erickson T., Smith D. N., Kellogg W. A., Laff M., Richards J. T. et Bradner E. (1999). *Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of "Babble"*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'99), 15-20 May 1999, Pittsburgh, Pennsylvanie, USA, pp. 72-79.
- [64] Faulring A. et Myers B. A. (2006). *Availability Bars for Calendar Scheduling*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'06), 22-27 April 2006, Montréal, Canada, pp. 760-765.

- [65] Fekete J.-D. (2004). *Elections 2004 : Données et Visualisation*. Université de Paris-Sud, Laboratoire de Recherche en Informatique, adresse Internet : <http://www.lri.fr/~fekete/elections2004/>, accédé le 9 août 2006.
- [66] Fisher D. et Dourish P. (2004). *Social and Temporal Structures in Everyday Collaboration*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'04), 24-29 Avril 2004, Vienne, Autriche, pp. 551-558.
- [67] Furnas, G. W. (1991). *The Fisheye Calendar System*. Rapport N° TM-ARH-020558, Bellcore Technical Memorandum, Novembre 1991.
- [68] *Gaim Multi-Protocol Instant Messaging Client*: adresse Internet: <http://gaim.sourceforge.net/about.php>, accédé le 2 août 2006.
- [69] George J. F. (2003). *Groupware*. Encyclopedia of Information Systems, Vol. 2, Elsevier, Academic Press, San Diego, pp. 509-518.
- [70] Gilbert E. et Karahalios K. (2006). *LifeSource: Two CVS Visualizations*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'06), 22-27 April 2006, Montréal, Canada, pp. 791-796.
- [71] Gräter W. et Prinz W. (2005). *Information Appliances Supporting Cooperation Awareness in Collaborative Business Processes*. Proceedings of the Conference on Collaborative Electronic Commerce Technology and Research (COLLECToR 2005), 12-13 Juin 2005, Furtwangen University, Allemagne.
- [72] Gray W.D. et Salzman M. (1998). *Damaged merchandise? A review of experiments that compare usability evaluation methods*. Human-Computer Interaction, Vol. 13, pp. 203-261.
- [73] Greenberg S., Gutwin C., Cockburn A. (1996). *Awareness Through Fisheye Views in Relaxed-WYSIWIS Groupware*. Proceedings of the Graphics Interface 1996 Conference, 22-24 Mai 1996, Toronto, Canada.
- [74] Gregory R. L. (2000). *L'oeil et le cerveau, la psychologie de la vision*, 5^{ème} Edition. De Boeck Université, Bruxelles.
- [75] Grinstein G., Keim D. et Ward M. (2004). *Information Visualization and Visual Data Mining*. Tutorial, 2004 IEEE Visualization Conference (Vis'2004), 10-15 octobre 2004, Austin, Texas, USA.
- [76] Grinstein G., Trutschl M. et Cvek U. (2001). *High-Dimensional Visualizations*. Proceedings of the 7th SIGKDD Workshop on Visual Data Mining (KDD-2001), San Francisco, Californie, USA, pp. 1-14.
- [77] Grinstein G. et Ward M. (2002). *Introduction to Data Visualization*. Fayyad U., Grinstein G., Wierse A. (Eds): Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann Publishers, Academic Press, San Diego, p. 29-30.
- [78] Grudin J. (1994). *CSCW: History and Focus*. IEEE Computer, Vol. 27, N° 5, Mai 1994, pp. 19-25.
- [79] Gruia-Catalin R. et Cox K. C. (1992). *Program Visualization: The Art of Mapping Programs to Pictures*. Proceedings of the 14th International Conference on Software Engineering, 1992, Melbourne, Australie, pp. 412-420.
- [80] Gutwin C. et Greenberg S. (1998). *Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness*. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'98), 14-18 Novembre 1998, Seattle, Washington, USA, pp.207-216.

- [81] Gutwin C. et Greenberg S. (1999), *A Framework of Awareness for Small Groups in Shared-Workspace Groupware*. Technical Report 99-1, Department of Computer Science, University of Saskatchewan, Canada.
- [82] Gutwin C. et Greenberg S. (1999). *The effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware*. ACM Transactions on Computer-Human Interaction, Vol. 6, N° 3, September 1999, pp. 243-281.
- [83] Gutwin C. et Greenberg S. (2002). *A Descriptive Framework of Workspace Awareness for Real-Time Groupware*. Journal of Computer Supported Cooperative Work, Vol. 11, pp. 411-446.
- [84] Gutwin C. et Greenberg S. (2004). *The Importance of Awareness for Team Cognition in Distributed Collaboration*. E. Salas and S.M. Fiore (Eds.) Team Cognition: Understanding the Factors that Drive Process and Performance, APA Press, Washington, 2004, pp. 177-201.
- [85] Gutwin C., Penner R. et Schneider K. (2004), *Group Awareness in Distributed Software Development*. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'04), 6-10 Novembre 2004, Chicago, Illinois, USA, pp.72-81.
- [86] Gwizdka J. (2002a). *Reinventing the Inbox – Supporting the Management of Pending Tasks in Email*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'02), 20-25 Avril 2002, Minneapolis, Minnesota, USA, pp. 550-551.
- [87] Gwizdka J. (2002b). *Future Time in Email – Design and Evaluation of a Task-based Email Interface*. Proceedings of the 2002 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON 2002), 30 Septembre-3 Octobre 2002, Toronto, Canada.
- [88] Hao J. J., Zhang K. et Huang M. *RELT – Visualizing Trees on Mobile Devices*. Proceedings of the International Conference on Visual Information Systems (VISUAL'07), 28-29 Juin 2007, Shanghai, Chine, pp. 344-357.
- [89] Logiciel HCIL 4.1.1. (2004), accessible à l'adresse: <http://www.cs.umd.edu/hcil/treemap/> , accédé le 18 juin 2008.
- [90] Heath C. et Luff P. (1991) *Collaborative activity and technological design: task coordination in London Underground control rooms*. Proceedings of ECSCW'91, 24-27 Septembre 1991, Amsterdam, Pays-Bas, pp. 65-80.
- [91] Hoffman P. et Grinstein G. (2002). *A Survey of Visualizations for High-Dimensional Data Mining*. Fayyad U., Grinstein G., Wierse A. (Eds): Information Visualization in data Mining and Knowledge Discovery, Morgan Kaufmann Publishers, Academic Press, San Diego, pp. 47-82.
- [92] Hollan J.D., Bederson B. B. et Helfman J. I. (1997), *Information Visualization*, Handbook of Human-Computer Interaction, 2^{ème} édition, Chap 2, pp 33-48.
- [93] Hollan J., Hutchins E. et Kirsh D. (2000), *Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research*. ACM Transactions on Computer-Human Interaction, Vol. 7, N° 2, Juin 2000, pp. 174-196.
- [94] Howard R. et Petersen E. (2001), *Monitoring communications in partnering projects*. Electronic Journal of Information Technology in Construction, Vol. 6, <http://itcon.org/2001/1>, accédé le 5 Août 2005.
- [95] Howell D. C. (1998). *Méthodes statistiques en sciences humaines*. De Boeck, Bruxelles. Traduction de *Statistical Methods for Psychology*, 4th Edition, Duxbury, Thomson Publishing Inc.
- [96] Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA, MIT Press.

- [97] Hutchins E. et Klausen T. (1996). *Distributed Cognition in an Airline Cockpit*. Communication and cognition at work, Middleton D., Engeström Y., Cambridge, Cambridge University Press, pp. 15-34.
- [98] Irani P., Slonowsky D. et Shajahan P. (2006). *Human perception of structure in shaded space-filling visualizations*. Information Visualization, Vol. 5, N° 1, pp. 47-61.
- [99] Johansen R. (1988). *Groupware: computer support for business teams*. The Free Press, New York, USA.
- [100] Johnson P. et Johnson H. (2001). *Interaction, Collaboration and Communication: A Human-Computer Interaction Perspective*. Accolade, Architecture, Collaboration, Design, M. Stellinwerff & J. Verbeke Eds, Delft University Press, Delft, pp. 43-57.
- [101] Johnson B. et Shneiderman B. (1991). *Treemaps: a space-filling approach to the visualization of hierarchical information structures*. Proceedings of the IEEE Conference on Visualization 1991 (San Diego, USA), Los Alamitos: IEEE Computer Society Press; 284-291.
- [102] Keim D. (2001). *Visual Exploration of Large Data Sets*. Communications of the ACM, Vol. 44, N° 8, Août 2001, pp. 38-44.
- [103] Keim D. (2002). *Information Visualization and Visual Data Mining*. IEEE Transactions on Visualization and Computer Graphics, Vol. 8, N° 1, pp. 1-8.
- [104] Kerr B. (2003). *Thread arcs: an email thread visualization*. IBM Research Report, RC22951 (W0310-175), Octobre 2003, accessible à l'adresse: <http://www.research.ibm.com/remail/publications.html>, accédé le 4 août 2005.
- [105] Kieras D. (2003). *Model-based evaluations*. Jacko J.A., Sears A. (Eds): The Human-Computer Interaction Handbook, Fundamentals, Evolving Technologies and Emerging Applications, Lawrence Erlbaum Associates Publishers, London, p. 1139-1151.
- [106] Kirakowski Jurek (2000). *Questionnaires in Usability Engineering*. Site web de l'auteur, accessible à l'adresse: <http://www.ucc.ie/hfrg/resources/>, accédé le 20 février 2007.
- [107] Kubicki S., Bignon J.-C., Halin G. et Humbert P. (2006). *Assistance to Building Construction Coordination – Towards a Multi-view Cooperative Platform*. Electronic Journal of Information Technology in Construction (ITCon), Vol. 11, Special issue on Process Modelling, Process Management and Collaboration, disponible à l'adresse <http://www.itcon.org/>.
- [108] Kurlander D., Skelly T. et Salesin D. (1996). *Comic Chat*. Proceedings of the 23rd Annual Conference on Computer graphics and Interactive Techniques (SIGGRAPH'96), 4-9 Août 1996, La Nouvelle Orleans, Louisiane, USA, pp. 225-236.
- [109] Lamping J. et Rao R. (1994). *Laying out and Visualizing Large Trees Using a Hyperbolic Space*. Proceedings of 7th annual ACM symposium on User interface software and technology (UIST'94), 2 – 4 novembre 1994, Marina del Rey, California, USA; pp. 13-14.
- [110] Larkin J. et Simon H. (1987). *Why a diagram is (sometimes) worth ten thousand words*. Cognitive science, Vol. 11, pp. 65-99.
- [111] Lee B., Parr C.S., Plaisant C., Bederson B., Veksler V.D., Gray W.D. et Kotfila C. (2006). *TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts*. IEEE Transactions on Visualization and Computer Graphics, Vol. 12, N° 6, pp. 1414-1426.
- [112] Leontiev A. (1978). *Activity, Consciousness, Personality*. Englewood Cliffs, NJ, Prentice Hall.
- [113] Leplat J. (1991). *Organization of activity in collective tasks*. Rasmussen J., Brehmer B. et Leplat J., Distributed decision making: cognitive models for cooperative work, John Wiley & Sons Ltd.

- [114] Lewis J. R. (1994). *Sample size for usability studies: Additional considerations*. Human Factors, Vol. 36, pp. 368-378.
- [115] Lewis J. R. (2006a). *Usability Testing*. Salvendy (Ed.): Handbook of Human Factors and Ergonomics, 3rd Edition, Wiley & Sons, New Jersey, pp. 1275-1316.
- [116] Lewis J. R. (2006b). *Sample Sizes for Usability Tests: Mostly Math, Not Magic*. Interactions, Special issue on Waits & Measures Quantifying Usability, Vol. 13, N° 6, pp. 29-33.
- [117] Wei-Jen L., Shlomo H. et Stolfo S. J. (2004). *Email Archive Analysis Through Graphical Visualization*. Proceedings of the VizSEC/DMSEC'04 Conference, 29 Octobre 2004, Washington, USA, pp. 128-132.
- [118] Linebarger J. M., Scholand A. J. et Ehlen M. A. (2006). *Representations and Metaphors for the Structure of Synchronous Multimedia Collaborations within Task-Oriented, Time-Constrained Distributed Teams*. Proceedings of 39th Hawaii International Conference on System Sciences (HICSS-39), 4-7 Janvier 2006, Kauai, Hawaii, USA.
- [119] Lohse J., Rueter H., Biolsi K. et Walker N. (1990). *Classifying Visual Knowledge Representations: A Foundation for Visualization Research*. Proceedings of the 1st IEEE Conference on Visualization, 23-26 Octobre 1990, San Francisco, Californie, USA, pp. 131-138.
- [120] Lohse G. L., Biolsi K., Walker N. et Rueter H. H. (1994). *A Classification of Visual Representations*. Communications of the ACM, Vol. 37, N° 12, Décembre 1994, pp. 36-49.
- [121] Lohse G. (1995). *Designing graphic decision aids: the role of working memory*. Working paper, Operations and Information Management Department, The Wharton School of the University of Pennsylvania.
- [122] Lohse G.L. (1997), *Models of Graphical Perception*. Handbook of Human-Computer Interaction, 2^{ème} édition, Elsevier, Amsterdam, pp 107-135.
- [123] Lonchamps J. (2003). *Le travail coopératif et ses technologies*. Editions Hermes, Paris.
- [124] Lottaz C., Stouffs R. et Smith I. (2000). *Increasing understanding during collaboration through advanced representations*. Electronic Journal of Information Technology in Construction, Vol. 5, 2000, <http://www.itcon.org/2000/1>, accessed 18 March 2005.
- [125] Mackay W. E. (1988). *More than just a communication system: diversity in the use of electronic mail*. Proceedings of the Conference on Computer-Supported Co-operative Work, 26-28 Septembre 1988, Portland, Oregon, USA, pp. 344-353.
- [126] Mackay W., Appert C., Beaudoin-Lafon M., Chapuis O., Du Y., Fekete J.-D. et Guiard Y. (2007). *Touchstone : Exploratory Design of Experiments*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'07), 28 Avril-3 Mai 2007, San Jose, Californie, USA, pp. 1425-1434.
- [127] Mackinlay J. D., Robertson G. G. et DeLine R. (1994). *Developing Calendar Visualizers for the Information Visualizer*. Proceedings of 7th Annual Symposium on User Interface Software and Technology (UIST'94), 2-4 Novembre 1994, Marina del Rey California, USA, pp. 109-117.
- [128] Malet R. (2004). *L'optique des jardins*. Editions Ulmer, Paris.
- [129] Malone T.W. et Crowston K. (1990). *What is Coordination Theory and How Can It Help Design Cooperative Work Systems*. Proceedings of the 3rd Conference on Computer-Supported Co-operative Work, 7-10 Octobre 1990, Los Angeles, Californie, USA, pp. 357-370.

- [130] Malone T.W. et Crowston K. (1994). *The interdisciplinary study of coordination*. ACM Computing Surveys, Vol. 26, N° 1, pp. 87-120.
- [131] Malone T.W., Crowston K., Lee J., Pentland B., Dellarocas C., Wyner G., Quimby J., Osborn C.S., Bernstein A., Herman G., Klein M. et O'Donnell E. (1999) *Tools for inventing organizations: Towards a handbook of organizational processes*. Management Science, Vol. 45, N° 3, pp. 425-443.
- [132] Masoodian M. et Budd D. (2004). *Visualization of Travel Information on PDAs*. Proceedings of the 5th Australasian User Interface Conference (AUIC 2004), 18-22 Janvier 2004, Dunedin, Nouvelle-Zélande, pp. 65-71.
- [133] Molich R., Thomsen A.D., Karyukina B., Schmidt L., Ede M., van Oel W. et Arcuri M. (1999). *Comparative evaluation of usability tests*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'99), 15-20 May 1999, Pittsburgh, Pennsylvanie, USA, pp. 83-84.
- [134] Mynatt E. et Tullio J. (2001). *Inferring Calendar Event Attendance*. Proceedings of the 6th international conference on Intelligent User Interfaces (IUI'01), 14-17 Janvier 2001, Santa Fe, Nouveau Mexique, USA, pp. 121-128.
- [135] Nardi B. A., Whittaker S., Isaacs E., Creech M., Johnson J. et Hainsworth J. (2002), *Integrating Communication and Information through ContactMap*. Communications of the ACM, Vol. 45 N° 4, pp. 89 - 95.
- [136] Newberry B. (2001). *Raising student social presence in online classes*. Proceedings of WebNet 2001 World Conference on the WWW and the Internet, 23-27 Octobre 2001, Orlando, Floride, USA, pp. 905-910.
- [137] Nielsen J. (1994) *Enhancing the explanatory power of usability heuristics*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'94), 24-28 Avril 1994, Boston, Massachusetts, USA, pp. 152-158.
- [138] Nielsen J. et Landauer T. K. (1993). *A Mathematical Model of the Finding of Usability Problems*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'93), 24-29 Avril 1993, Amsterdam, Pays-Bas, pp. 206-213.
- [139] Noirhomme-Fraiture M. (2005). *Statistiques pour l'IHM*. Tutorial dispensé lors de la 17^{ème} Conférence Francophone sur l'Interaction Homme-Machine (IHM 2005), 27 – 30 Septembre 2005, Toulouse, France.
- [140] Noirhomme-Fraiture M. et Serpe V. (1998), *Visual representation of hypermedia links according to their types*. Proceedings of the working conference on Advanced Visual Interfaces (AVI'98), 24-27 Mai 1998, L'Aquila, Italie, pp. 146-155.
- [141] Norman D. (1993). *Things That Make Us Smart*. Reading, MA: Addison-Wesley.
- [142] Novick L.R. et Hurley S. M. (2001), *To matrix, Network, or Hierarchy: That is the Question*. Cognitive Psychology, Vol. 42, pp. 158-216.
- [143] O'Donnel R., Dix A. et Ball L. J. (2006) *Exploring the PieTree for Representing Numerical Hierarchical Data*. Proceedings of the 20th Conference of the Human-Computer Interaction Group of the British Computer Society (HCI 2006), 11-15 Septembre 2006, London, UK.
- [144] Olson J. S. et Olson G. M. (2003). *Computer-Supported Cooperative Work*. Encyclopedia of Information Systems, Vol. 1, Elsevier, Academic Press, San Diego, pp. 243-253.
- [145] Otjacques B. et Feltz F. (2004), *VisuThread: un module de visualisation de conversations électroniques*. Actes de la 16^{ème} Conférence Francophone sur l'Interaction Homme-Machine (IHM 2004), (16th French-speaking Conference on HCI), 30 Août – 3 Septembre 2004, Namur, Belgique, pp. 239-242.

- [146] Otjacques B., Feltz F., Halin G. et Bignon J.-C. (2005), *Mat'Graph : Transformation matricielle de graphe pour visualiser des échanges électroniques*. Actes de la 17^{ème} Conférence Francophone sur l'Interaction Homme-Machine (IHM 2005), (17th French-speaking Conference on HCI), 27 – 30 Septembre 2005, Toulouse, France.
- [147] *The Palace User Software Guide for Windows*, version 3.5, <http://www.palacechat.us/documentation.php>, accédé le 7 Juin 2005.
- [148] Palmer S. E. (1992). *Common region: A new principle of perceptual grouping*. Cognitive Psychology 1992: 24: 436-447.
- [149] Palmer S. E. et Rock I. (1994). *Rethinking perceptual organization: The role of uniform connectedness*. Psychonomic Bulletin and Review 1994 1(1), pp. 29-55.
- [150] Pavard B., Bressole M.C. et Evrard P. (1994). *Modélisation du dialogue coopératif à partir de la théorie de la pertinence et des règles de coopération*. Rapport de recherche GRIC-ARAMMIHS 94/07-001.
- [151] Perer A. et Smith M. A. (2006). *Contrasting Portraits of Email Practices: Visual approaches to reflection and analysis*. Proceedings of the working conference on Advanced Visual Interfaces (AVI'06), 23-26 Mai 2006, Venise, Italie, pp. 389-395.
- [152] Perry M. (1997). *Distributed cognition and computer supported collaborative design : the organisation of work in construction engineering*. Thèse de doctorat, Université de Brunel, UK.
- [153] Persson P. (2003). *ExMS: an Animated and Avatar-based Messaging System for Expressive Peer Communication*. Proceedings of the International Conference on Supporting Group Work (GROUP'03), 9-12 Novembre 2003, Sanibel Island, Floride, USA, pp. 31-39.
- [154] Pfitzner D., Hobbs V. et Powers D. (2003). *A Unified Taxonomic Framework for Information Visualization*. Proceedings of the Australian Symposium on Information Visualization 2003, Adelaïde, Australia, pp. 57-66.
- [155] Pinho, M.S., Bowman, D.A., and Freitas C.M.D.S. *Cooperative Object Manipulation in Immersive Virtual Environments: Framework and Techniques*. Proceedings of the ACM symposium on Virtual Reality Software and Technology (VRST'02), 11-13 November 2002, Hong-Kong, Chine, pp. 171-178.
- [156] Plaisant C., Milash B., Rose A., Widoff S. et Shneiderman B. (1996). *Lifelines: visualizing personal histories*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'96), 13-18 Avril 1996, Vancouver, Canada, pp. 221-227.
- [157] Prinz W., Kolvenbach S., et Klöckner K. *Situative Cooperation Support for Communities*. SIGGROUP Bulletin, Décembre 2002, Vol. 23(3), pp. 24-28.
- [158] Prinz W., Pankoke-Babatz U., Gräther W., Gross T., Kolvenbach S. et Schäfer L. (2004). *Presenting Activity Information in an Inhabited Information Space*. Snodown, Chirchill et Frécon (Eds): *Inhabited Information Spaces*, CSCW Series, Springer, Londres, pp.181-208.
- [159] QUIS Site web, accessible à l'adresse: <http://www.lap.umd.edu/QUIS/index.html>, accédé le 19 février 2007.
- [160] Riendeau S. (2006). *Space-Time Navigator for Architectural Projects*. Electronic Journal of Information Technology in construction (Itcon), Vol. 11, pp. 1-15, accessible à l'adresse Internet : <http://www.itcon.org/2006/1>, accédé le 24 juillet 2006.
- [161] Roberston G., Card S.K. et Mackinlay J.D. (1993). *Information visualization using 3D interactive animation*. Communication of the ACM, Vol. 36 N° 4, pp. 57-71.

- [162] Robertson P. K. (1990). *A Methodology for Scientific Data Visualisation: Choosing Representations Based on a Natural Scene Paradigm*. Proceedings of the 1st IEEE Conference on Visualization, 23 – 26 Octobre 1990, San Francisco, Californie, USA, pp. 114-123.
- [163] Robertson G. G., Czerwinski M. P. et Churchill J.E. (2005). *Visualization of Mapping Between Schemas*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'05), 2-7 Avril 2005, Portland, Oregon, USA, pp. 431-439.
- [164] Robertson G. G., Mackinlay J.D. et Card S.K. (1991). *Cone Trees : Animated 3D Visualisations of Hierarchical Information*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'91), 27 Avril – 2 Mai 1991, New Orleans, Louisiana, USA, pp. 189-194.
- [165] Robinson M., Pekkola S., Korhonen J., Hujala S., Toivonen T. et Saarinen M.-J. O. (2001). *Extending the Limits of Collaborative Virtual Environments*. Churchill E. F., Snowdon D. N. et Munro A. J. (Eds): Collaborative Virtual Environments, CSCW Series, Springer, pp. 21-42.
- [166] Rodden T. (1996). *Populating the Application: a Model of Awareness for Cooperative Applications*. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'96), 16-20 Novembre 1996, Boston, USA, pp. 87-96.
- [167] Rodenstein R. et Donath J. (2000). *Talking in Circles: Designing a Spatially-Grounded Audioconferencing Environment*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'00), 01-06 Avril 2000, The Hague, Pays-Bas, pp. 81-88.
- [168] Rodrigues J.F., Traina A.J.M., de Oliveira M.C.F. et Traina C. (2006), *Reviewing Data Visualization : an Analytical Taxonomical Study*. Proceedings of Information Visualization (IV'06), Juillet 2006, Londres, Royaume-Uni, pp. 713-720.
- [169] Rognin L. (1997), *Coordination Theory*. Working Paper, COTCOS project, accessible à l'adresse Internet: [http:// www-sv.cict.fr/cotcos/pjs/](http://www-sv.cict.fr/cotcos/pjs/), accédé le 15 septembre 2005.
- [170] Rohall S. L., Gruen D., Moody P. et Kellerman S. (2001), *E-mail visualizations to aid communication*. Proceedings of the IEEE Symposium on Information Visualization, 22-23 Octobre 2001, San Diego, Californie.
- [171] Rosenberger S. T., VanKleeck M., Vicente A. et Smith B. K. (2000). *Fugue: A Computer Mediated Conversational System that Supports Turn Negotiation*. Proceedings of 33th Hawaii International Conference on System Sciences (HICSS-33), 4-7 Janvier 2000, Island of Maui, Hawaii, USA.
- [172] Roth S. F. et Mattis J. (1990). *Data Characterization for Intelligent Graphics Presentation*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'90), Avril 1990, Seattle, Washington, USA, pp. 193-200.
- [173] Saltz J.S., Hiltz S.R. et Turoff M. *Student Social Graphs: Visualizing a Student's Online Social Network*. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'04), 6-10 Novembre 2004, Chicago, Illinois, USA, pp. 596-599
- [174] Saporta G. (2006). *Probabilités, analyse des données et statistique* (2^{ème} Edition). Editions Technip, Paris.
- [175] Sears A. (1997). *Heuristic Walkthroughs: Finding the problems without the noise*. International Journal of Human-Computer Interaction, Vol. 9, pp. 213-234.
- [176] Sears A. (2003). *Testing and Evaluation*. Jacko J.A., Sears A. (Eds): The Human-Computer Interaction Handbook, Fundamentals, Evolving Technologies and Emerging Applications, Lawrence Erlbaum Associates Publishers, London, p. 1091-1092.
- [177] Segal L. (1994). *Effects of Checklist Interface on Non-Verbal Crew Communications*. NASA Ames Research Center, Contractor Report 177639.

- [178] Seifert C.M. et Hutchins E. (1992). *Error as opportunity: learning in a cooperative task*. Human-Computer Interaction, 7, 1992, pp. 409-435.
- [179] Shneiderman B. (1992) *Tree Visualization with Tree-Maps: 2-d Space-Filling Approach*. ACM Transactions on Graphics, 1992 11(1), pp. 92-99.
- [180] Shneiderman B. (1996). *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. Proceedings of the IEEE Symposium on Visual Languages, Septembre 1996, Boulder, Colorado, USA, pp. 336-343.
- [181] Shneiderman B. (1998), *Designing the User Interface*, 3ème Edition. International Edition, Addison Wesley, Reading, USA, pp. 510-541.
- [182] Shneiderman B. (2006). *Treemaps for space-constrained visualization of hierarchies*. University of Maryland, Human – Computer Interaction Lab, adresse Internet: <http://www.cs.umd.edu/hcil/treemap-history/>, accédé le 9 août 2006.
- [183] Shneiderman B. et Plaisant C. (2005), *Designing the User Interface*, 4ème Edition. International Edition, Addison Wesley, Boston, USA, pp. 580-590.
- [184] Simon H. (1969). *The Sciences of the Artificial*. MIT Press.
- [185] Smartmoney.com, adresse Internet : <http://www.smartmoney.com>, accédé le 11 décembre 2006.
- [186] Smith M. A., Farnham S. D. et Drucker S. M. (2000). *The Social Life of Small Graphical Chat Spaces*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'2000), 1-6 Avril 2000, La Haye, Pays-Bas, pp. 462-469.
- [187] Smith M. A. et Fiore A. T. (2001). *Visualisation Composants for persistent conversations*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'01), 31 Mars - 5 Avril 2001, Seattle, Washington, USA, pp. 136-143.
- [188] Smith S. et Mosier J. (1986). *Guidelines for designing user interface software*. Report ESD-TR-86-278. The MITRE Corporation, Bedford, MA, disponible à l'adresse Internet: <ftp://ftp.cis.ohio-state.edu/pub/hci/Guidelines/>.
- [189] Sociable Media Group, Massachusetts Institute of Technology, adresse Internet: <http://smg.media.mit.edu/projects.html>, accédé le 13 juillet 2006.
- [190] Spence R. (2007). *Information Visualization – Design for Interaction*, 2nd Edition. Pearson Prentice Hall, Harlow.
- [191] Sperber D. et Wilson D. (1987). *La pertinence*. Editions Minuit, Paris.
- [192] SpiraClock (site web), adresse Internet : <http://www.emn.fr/x-info/spiraclock/>, accédé le 26 juillet 2006.
- [193] Spool J et Schroeder W. (2001). *Test web sites: five users is nowhere near enough*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'01), 31 Mars - 5 Avril 2001, Seattle, Washington, USA, pp. 285-286.
- [194] St Ger B. and Engeli M. (1998). *Information Technology for Visual Enterprises Meta-visualizations of Document-Structures*. ETH, Swiss Federal Institute of Technology, Architecture & CAAD, disponible à l'adresse Internet: <http://e-collection.ethbib.ethz.ch>, accédé le 4 Août 2005.
- [195] SUMI web site, disponible à l'adresse: <http://sumi.ucc.ie>, accédé le 20 février 2007.
- [196] Tat A. et Carpendale S. (2006). *CrystalChat : Visualizing Chat History*. Proceedings of 39th Hawaii International Conference on System Sciences (HICSS-39), 4-7 Janvier 2006, Kauai, Hawaii, USA.
- [197] Tilley S. et Huang S. (2002). *Documenting Software Systems with Views III: Towards a Task-Oriented Classification of Program Visualization Techniques*. Proceedings of the 20th

Annual International Conference on Computer Documentation (SIGDOC'02), 20-23 Octobre 2002, Toronto, Canada, pp. 226-233.

[198] Tory M. et Möller T. (2002). *A Model-Based Visualization Taxonomy*. Technical Report CMPT-TR2002-06, Simon Fraser University.

[199] Tufte E. R. (2001). *The Visual Display of Quantitative Information*, 2^{ème} Edition. Graphics Press, Cheshire, Connecticut, USA, p 13-15, 92. (1^{ère} Edition publiée en 1992)

[200] Tweedie L. (1997). *Characterizing Interactive Externalizations*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'97), Mars 1997, Atlanta, Georgia, USA, pp. 375-382.

[201] Umanath N. et Scamell R. (1988). *An experimental investigation of the impact of data display format on recall performance*. Communications of the ACM, Vol. 31, pp.562-570.

[202] Unwin A., Theus M. et Härdle W.K. (2008). *Exploratory Graphics of a Financial Dataset*. Chen, C., Härdle W., Unwin (Eds): Handbook of Data Visualization, Springer, 2008, pp. 831-852.

[203] Van Hentenryck P. (1989). *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, Massachusetts, USA.

[204] Venolia G. D. et Neustaedter C. (2003), *Understanding Sequence and Reply Relationships within Email Conversations: A Mixed-Model Visualization*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'03), 5-10 Avril 2003, Ft. Lauderdale, Floride, USA, pp. 361-368.

[205] Viégas F. B., Golder S. et Donath J. (2006). *Visualizing Email Content: Protraying Relationships from Conversational Histories*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'06), 22-27 Avril 2006, Montréal, Canada, pp. 979-988.

[206] Viégas F. B. et Smith M. (2004). *Newsgroup Crowds and AuthorLines: Visualizing the Activity of Individuals in Conversational Cyberspaces*. Proceedings of 37th Hawaii International Conference on System Sciences (HICSS-37), 5-8 Janvier 2004, Hilton Waikoloa Village, Island of Hawaii, Hawaii, USA.

[207] Viégas F. B., Wattenberg M. et Dave K. (2004), *Studying Cooperation and Conflict between Authors with history flow Visualizations*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'04), 24-29 Avril 2004, Vienne, Autriche, pp. 575-582.

[208] Virzi R.A. (1990). *Streamlining the design process: Running fewer subjects*. Proceedings of the 34th Annual Meeting of the Human Factors Society, Santa Monica, USA, pp. 291-294.

[209] Virzi R.A. (1992). *Refining the test phase of usability evaluation: How many subjects is enough?* Human Factors, Vol. 34, pp. 457-468.

[210] Vliegen R., van Wijk J.J. et van der Linden E.-J. (2006) *Visualizing Business Data with Generalized Treemaps*. IEEE Transactions on Visualization and Computer Graphics, Vol. 12, N°5, Sept./Oct. 2006, pp. 789-796.

[211] Vygotsky L. (1978). *Mind and society: the development of higher psychological processes*. Cambridge, MA, Harvard University Press.

[212] Ware C. (2004). *Information Visualization, Perception for Design*. San Francisco, CA, Morgan Kaufmann Publishers.

[213] Ware C. (2003). *Design as Applied Perception*. Dans J.M. Carroll (Eds): HCI Models, Theories, and Frameworks, Towards a multidisciplinary science, Morgan Kaufmann Publishers, San Francisco, p. 11-26.

[214] Wasserman S. et Faust K. (1999), *Social Network Analysis, Methods and Applications*. Cambridge University Press, édition de 1999 (première edition en 1994).

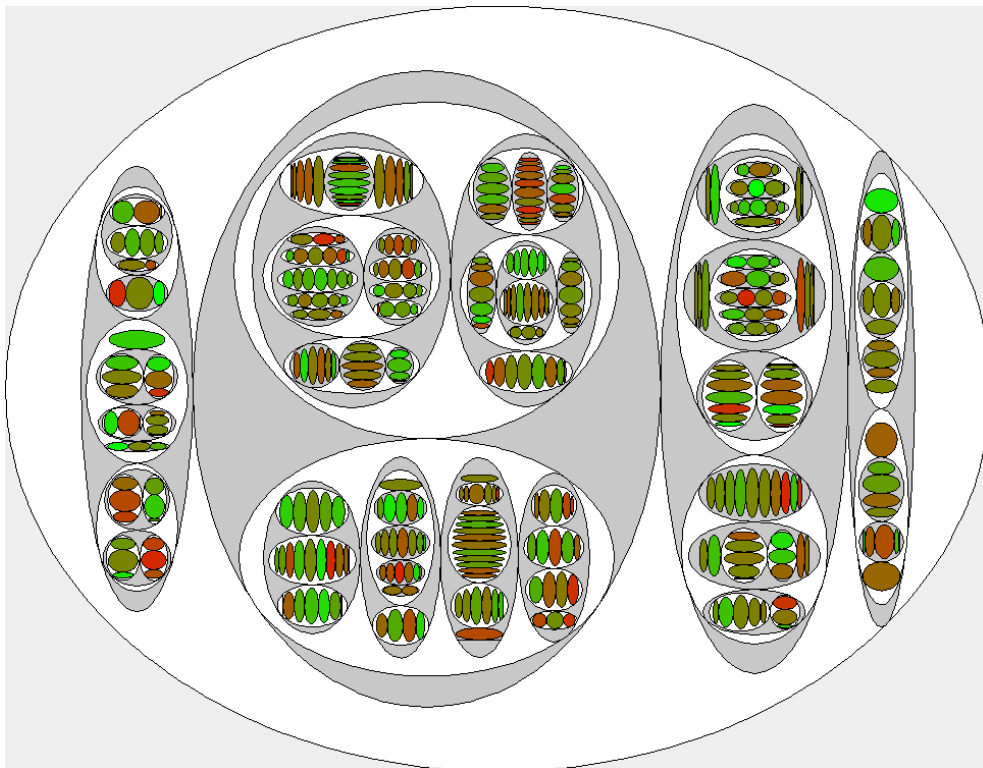
- [215] Wehrend S. et Lewis C. (1990). *A Problem-oriented Classification of Visualization Techniques*. Proceedings of the 1st IEEE Conference on Visualization, 23-26 Octobre 1990, San Francisco, Californie, USA, pp. 139-143.
- [216] Wenzel S., Bernhard J. et Jessen U. (2003). *A Taxonomy of Visualization Techniques for Simulation in Production and Logistics*. Proceedings of the 2003 Winter Simulation Conference, 7-10 Décembre 2003, New Orleans, Louisiane, USA, pp. 729-736.
- [217] Weskamp M. (2004). *Newsmap*. Adresse Internet: <http://www.marumushi.com/apps/newsmap/>, accédé le 9 août 2006.
- [218] Wetzel K. (2004). *Using Circular Treemaps to visualize disk usage*. Adresse Internet: <http://lip.sourceforge.net/ctreemap.html>, accédé le 9 août 2006.
- [219] Whittaker S. et Sidner C. (1996). *Email overload: exploring personal information management of email*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'96), 13-18 Avril 2004, Vancouver, Canada, pp. 276-283.
- [220] Wikipedia (2006). Adresse Internet : <http://www.wikipedia.org>
- [221] Winckler M.A., Palanque P. et Freitas C.M.D.S. (2004). *Tasks ans Scenario-based Evaluation of Information Visualization Techniques*. Proceedings of 3rd International Workshop on Task Models and Diagrams for User Interface Design (TAMODIA'04), 15-16 Novembre 2004, Prague, Rép. Tchèque, pp. 165-172.
- [222] Woolrych A. et Cockton G. (2003). *The "Magic Number 5": Is it Enough for Web Testing?* Proceedings of the Conference on Human Factors in Computing Systems (CHI'03), 5-10 Avril 2003, Ft. Lauderdale, Floride, USA, pp. 698-699.
- [223] van Wijk JJ. et van de Wetering H. (1999). *Cushion TreeMaps*. Proceedings of the IEEE Symposium on Information Visualization (InfoVis 1999), 24-29 Octobre 1999, San Francisco, USA, pp. 73-78.
- [224] Yiu K. S., Baecker R., Silver N. et Long B. (1997), *A Time-based interface for electronic mail and task management*. Design of Computing Systems: Proceedings of HCI International'97. Vol. 2. Elsevier, pp. 19-22.
- [225] Zhang J. et Norman D.A., *Representation in distributed cognition tasks*. Cognitive Science, Vol. 18, 1994, pp. 87-122.
- [226] Zizi M. et Beaudoin-Lafon M., *Accessing Hyperdocuments through Interactive Dynamic Maps*. Proceedings of the ACM European Conference on Hypermedia Technology (ECHT'94), 19-23 Septembre 1994, Edinbourg, Ecosse, pp. 126-135.

CHAPITRE 1 : INTRODUCTION	5
1. REMERCIEMENTS	5
2. MOTIVATION DES TRAVAUX.....	6
3. DEMARCHE GLOBALE	7
4. CONTRIBUTIONS MAJEURES DE LA THESE	9
5. PLAN	12
CHAPITRE 2 : LE SUPPORT INFORMATISÉ À LA COOPÉRATION.....	13
1. LES THÉORIES DE L'ACTIVITÉ COOPÉRATIVE	13
1.1 <i>La théorie de l'activité</i>	13
1.2 <i>La cognition distribuée</i>	14
1.3 <i>La théorie de la coordination</i>	16
1.4 <i>L'ergonomie cognitive</i>	17
2. LE TRAVAIL COOPÉRATIF ASSISTÉ PAR ORDINATEUR.....	18
2.1 <i>Définition</i>	18
2.2 <i>Modèle des 3C (Ellis)</i>	19
2.3 <i>Typologie des outils de support au travail coopératif</i>	19
3. LE CONCEPT DE CONSCIENCE DE GROUPE	21
3.1 <i>Définitions</i>	21
3.2 <i>Paradigmes de mise en œuvre</i>	22
3.2.1. <i>Modèle Publication / Souscription</i>	22
3.2.2. <i>Modèle spatial</i>	23
3.2.3. <i>Modèle de traitement de la connaissance de Brézillon et al.</i>	24
3.3 <i>Workspace Awareness Framework</i>	25
3.3.1. <i>Contexte</i>	25
3.3.2. <i>Description</i>	26
4. DISCUSSION	29
CHAPITRE 3 : VISUALISATION D'INFORMATION	35
1. INTRODUCTION GÉNÉRALE	35
2. TAXONOMIES DES TECHNIQUES DE VISUALISATION.....	39
2.1 <i>Perspective historique</i>	39
2.2 <i>Taxonomies proposées dans la littérature</i>	39
2.2.1. Bertin (1973, 1998 & 2000)	39
2.2.2. Roth et Mattis (1990).....	42
2.2.3. Robertson (1990)	44
2.2.4. Wehrend & Lewis (1990)	45
2.2.5. Tufte (1992).....	47
2.2.6. Gruia-Catalin & Cox (1992).....	47
2.2.7. Lohse, Biolsi, Walker & Rueter (1990 & 1994).....	49
2.2.8. Shneiderman (1996)	52
2.2.9. Card & Mackinlay (1997).....	53
2.2.10. Tweedie (1997).....	55
2.2.11. Chi (2000).....	57
2.2.12. Grinstein et al. (2001 & 2002).....	58
2.2.13. Tory & Möller (2002).....	62
2.2.14. Tilley & Huang (2002)	64
2.2.15. Pfitzner, Hobbs & Powers (2003).....	65
2.2.16. Wenzel, Bernhard & Jessen (2003)	67
2.2.17. Grinstein, Keim & Ward (2001, 2002, 2004).....	71
2.2.18. Rodrigues, Traina, de Oliveira et Traina (2006).....	73
3. MÉTA-TAXONOMIE DES TECHNIQUES DE VISUALISATION.....	75
3.1 <i>Introduction</i>	75

3.2	<i>Principes constructifs d'une taxonomie.....</i>	75
3.3	<i>Propriétés des taxonomies proposées dans la littérature.....</i>	77
3.3.1.	Domaine de définition	77
3.3.2.	Critères de différenciation des visualisations	79
3.3.3.	Remarque sur la démarche de construction des taxonomies	82
3.3.4.	Propriétés des critères de différenciation.....	83
3.3.5.	Qualités d'une bonne taxonomie	85
CHAPITRE 4 : VISUALISATION ET COOPÉRATION.....		87
1.	VISUALISATIONS DE COMMUNICATIONS	87
1.1	<i>Communications par messagerie électronique.....</i>	87
1.2	<i>Communications via chat et messagerie instantanée</i>	95
1.3	<i>Groupes de discussion, forums.....</i>	101
2.	VISUALISATIONS D' ACTIONS SUR DES RESSOURCES PARTAGÉES.....	103
3.	VISUALISATIONS DE LA PRÉSENCE DANS UN ENVIRONNEMENT PARTAGÉ	109
4.	VISUALISATIONS D'ÉVÉNEMENTS À CARACTÈRE COLLECTIF.....	112
5.	VISUALISATIONS COMBINÉES	118
6.	CONCLUSIONS.....	122
CHAPITRE 5 : TAXONOMIE DES TECHNIQUES DE VISUALISATIONS DE LA COOPÉRATION		125
1.	CADRE DE TRAVAIL	125
2.	APPROCHE GÉNÉRALE	125
2.1	<i>Définitions</i>	125
2.2	<i>Méthodologie de conception de la taxonomie IVF.....</i>	126
2.3	<i>Principes directeurs de la taxonomie IVF.....</i>	127
2.4	<i>Notations.....</i>	128
3.	LA TAXONOMIE IVF	129
3.1	<i>Propriétés descriptives de la situation réelle</i>	129
3.1.1.	Propriétés descriptives des entités réelles.....	129
3.1.2.	Propriétés descriptives des interactions réelles.....	130
3.2	<i>Propriétés descriptives du modèle du monde réel.....</i>	132
3.2.1.	Propriétés descriptives des entités du modèle	132
3.2.2.	Propriétés descriptives des interactions du modèle	134
3.3	<i>Propriétés descriptives de la technique de visualisation.....</i>	137
3.3.1.	Propriétés descriptives de la technique de visualisation en général	137
3.3.2.	Propriétés descriptives du contexte d'utilisation de la visualisation	138
3.3.3.	Propriétés descriptives de la représentation des entités	140
3.3.4.	Propriétés descriptives de la représentation des interactions	142
3.4	<i>Vue synoptique de la taxonomie IVF.....</i>	145
CHAPITRE 6 : INDICES D'ACTIVITÉ ET INTERACTIONS		149
1.	INTRODUCTION	149
2.	CONSCIENCE DE GROUPE	149
2.1	<i>Travaux antérieurs</i>	149
2.2	<i>Développements théoriques sur le concept.....</i>	150
3.	METRIQUES D'ACTIVITES COOPERATIVES	152
3.1	<i>Travaux antérieurs</i>	152
3.2	<i>Proposition de métriques.....</i>	153
3.2.1.	Motivation et identification des métriques	153
3.2.2.	Comparabilité des métriques : le problème du temps.....	154
3.2.3.	Contraintes de base.....	155
3.2.4.	Indice Coopadex – Qt.....	156
3.2.5.	Indices Coopidex – Qual et N-Qual	157
3.2.6.	Indice Glocoopex – G.....	159

3.2.7.	Remarque finale sur la portée du modèle de métrique	163
3.2.8.	Illustration des indices avec des données simulées	163
4.	VISUALISATION DES INDICES D'ACTIVITÉ	164
4.1	<i>Approche technologique</i>	164
4.2	<i>Qualification des interactions</i>	165
4.3	<i>Evolution de l'activité coopérative</i>	166
4.4	<i>Niveau d'activité courant</i>	168
5.	VISUALISATION 'CLOCKVIEW'	172
5.1	<i>Contexte de travail</i>	172
5.2	<i>Principes de conception</i>	172
5.3	<i>Prototype</i>	174
5.3.1.	Architecture logicielle	174
5.3.2.	Interface utilisateur et fonctionnalités	175
CHAPITRE 7 : ELLIMAPS ET DÉRIVÉS		179
1.	CONTEXTE	179
2.	VISUALISATION DE DONNÉES HIÉRARCHIQUES	180
3.	ELLIMAPS : VERSION DE BASE	189
3.1	<i>Principes constructifs</i>	189
3.2	<i>Choix de l'axe de division inter-niveaux</i>	193
4.	ELLIMAPS CIRCULARISÉES	195
5.	OPTIMISATION DE L'ALGORITHME	198
5.1	<i>Lacunes observées</i>	198
5.1.1.	Utilisation de l'espace d'affichage	198
5.1.2.	Effet de distorsion inter-niveaux	200
5.2	<i>Algorithme d'optimisation DGC</i>	202
5.2.1.	Problématique des nœuds cousins	217
5.3	<i>Analyse de l'avantage de l'algorithme optimisé</i>	218
5.4	<i>Représentation du nœud racine de la hiérarchie</i>	221
6.	ELLIMAPS-RÉSERVOIRS	222
6.1	<i>Problématique</i>	222
6.2	<i>Le concept de jauge</i>	222
6.3	<i>La transformation de formes géométriques en jauges</i>	225
6.4	<i>L'ajout de jauges à des techniques existantes</i>	225
6.4.1.	Ajout de jauges à un treemap	226
6.4.2.	Ajout de jauges à un treemap circulaire	226
6.4.3.	Ajout de jauges à un bubble tree	227
6.4.4.	Ajout de jauges à une ellimap	227
6.5	<i>Le principe des ellimaps-réservoirs</i>	228
6.6	<i>Développements géométriques</i>	229
CHAPITRE 8 : ELLIMAPS : EVALUATION		237
1.	LA PROBLÉMATIQUE DE L'ÉVALUATION	237
1.1.1.	Cadre général	237
2.	PLAN D'ÉVALUATION	243
3.	EVALUATION FORMATIVE	245
3.1	<i>Test F1</i>	245
3.1.1.	Méthodologie	245
3.1.2.	Résultats	245
3.2	<i>Tests F2</i>	246
3.2.1.	Méthodologie	246
3.2.2.	Résultats	247
3.3	<i>Tests F3i</i>	248
3.3.1.	Méthodologie	248
3.3.2.	Résultats	248

3.4	<i>Test F4</i>	249
3.4.1.	Méthodologie.....	249
3.4.2.	Résultats	250
4.	EVALUATION SOMMATIVE.....	252
4.1	<i>Méthodologie</i>	252
4.1.1.	Vue d'ensemble	252
4.1.2.	Techniques de visualisation évaluées	253
4.1.3.	Population.....	254
4.1.4.	Choix des observations.....	255
4.1.5.	Plan d'expérience	262
4.2	<i>Mise en œuvre du plan d'expérience</i>	268
4.2.1.	Expériences pilotes.....	268
4.2.2.	Déroulement des tests réels	269
4.2.3.	Infrastructure technique de base	272
4.2.4.	Environnement spécifique de test : FESTIV	274
4.3	<i>Résultats</i>	278
4.3.1.	Taux d'erreur et temps de réponse	278
4.3.2.	Evaluation subjective.....	286
4.3.3.	Conclusions	289
CHAPITRE 9 : PROTOTYPE 'ELLIMAPS'		291
1.	FONCTIONNALITÉS.....	291
1.1	<i>Organisation générale</i>	291
1.2	<i>Interaction avec les données</i>	292
1.3	<i>Algorithmes de positionnement des nœuds</i>	295
1.4	<i>Tri</i>	298
1.5	<i>Politique de coloriage</i>	299
1.6	<i>Ellimap-réservoirs</i>	302
1.7	<i>Relation entre données et variables graphiques</i>	304
2.	UTILISATION POUR VISUALISER LA COOPÉRATION.....	305
CHAPITRE 10 : CONCLUSIONS ET PERSPECTIVES		311
1.	TAXONOMIES	311
2.	ACTIVITÉ COOPÉRATIVE	312
3.	ELLIMAPS.....	312
4.	PERSPECTIVES.....	313
5.	POSTFACE	316
CHAPITRE 11 : BIBLIOGRAPHIE		319



Représentation d'une structure hiérarchique par ellimap