

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

De l'analyse organisationnelle à l'analyse des exigences

Clause, Olivier

Award date:
2002

[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, Namur

Institut d'informatique

Année académique 2001-2002

**De l'analyse organisationnelle
à l'analyse des exigences**

Olivier Clause

Mémoire en vue de l'obtention du grade de
maître et licencié en informatique

CBS 10076850

Remerciements

Je désire remercier le professeur Claire Lobet-Maris pour le temps et la patience qu'il m'a consacrés afin de réaliser ce mémoire et la connaissance de qualité qu'elle m'a transmise.

Je remercie également Michaël Petit et Anne Rousseau pour leur contribution à la rédaction de ce mémoire.

Je remercie tout particulièrement le docteur Neil Maiden ainsi que toute son équipe qui m'ont accueilli à bras ouverts au sein du HCID de la City University à Londres. Un grand merci pour l'expérience et le savoir que j'ai partagés et l'excellent contact que j'ai eu avec eux.

Je tiens aussi à remercier Pete Pavan, Marina Krumbholz, Raquel Monja, Ursula Armitage et Alexis Gizikis qui ont contribué à rendre mon séjour à Londres plus agréable.

Je remercie aussi vivement Vincent Letocart pour son aide précieuse et ses conseils avisés pour l'utilisation de $\text{\LaTeX}$

Merci à mes parents et à mes amis qui m'ont soutenu moralement pendant la rédaction de ce mémoire.

Résumé

L'informatique étant présent de plus en plus dans la vie courante il est devenu important de bien définir et de bien modéliser à la base le système d'information. Dans le cycle de vie d'un projet, deux étapes sont fondamentales à la réussite de celui-ci, à savoir l'analyse organisationnelle de l'existante et partant de cette analyse la modélisation des exigences du nouveau système. Un modèle pouvant intégrer l'analyse organisationnelle est le modèle i^* .

Le but de mon stage à Londres a été de bien comprendre le modèle i^* et à essayer de l'améliorer. Un fois ceci réalisé, j'ai pu me consacrer à la réalisation de mon mémoire qui a consisté à essayer de faire le rapprochement entre les théories des organisations et le modèle i^* . Ce mémoire propose donc une analyse du modèle i^* permettant de faire le lien avec l'analyse organisationnelle et une évolution possible du modèle pour intégrer les concepts organisationnels qui ne sont pas modélisables.

Abstract

Data processing being present more and more in the common life it became important to define well and to model well the information system. Two stages are fundamental in the success of this one, namely the organizational analysis of the existing and with this analysis the modelling of the requirements of the new system. A model being able to integrate the organizational analysis is the i^* model.

The goal of my visit in London was to understand deeply the model i^* and try to improve it. When I came back in Belgium, I was able to work myself to the realization of my master thesis which consisted of trying to make the link between the theories of organizations and the i^* model. This thesis proposes an analysis of the i^* model allowing to make the link with the organizational analysis. I also suggest a possible evolution of the actual model to integrating the organizational concepts that are not possible to modelize with this model.

Table des matières

Introduction	13
1 Le modèle i^*	17
1.1 Motivation	17
1.2 The Strategic Dependency Model	18
1.3 The Strategic Rationale Model	18
1.4 Travaux accomplis avec le modèle i^*	19
1.5 Autres recherches déjà accomplies dans ce domaine	20
1.5.1 La méthode <i>KAOS</i>	21
1.5.2 La méthode NFR	22
2 Les théories des organisations	23
2.1 L'acteur et sa stratégie	23
2.2 Le pouvoir comme fondement de l'action organisée	25
2.2.1 Définition	25
2.2.2 Les sources de pouvoir	26
2.2.3 Conclusion	27
2.3 Les buts dans une organisation	28
2.3.1 Notion	28
2.3.2 Les rapports entre buts	30
2.3.3 Buts spécifiques et buts organisationnels	32
3 Les éléments du modèle i^*	33
3.1 Acteurs	33
3.1.1 Représentation	33
3.2 Éléments associés avec les acteurs	34
3.2.1 Goal	34
3.2.2 Task	34
3.2.3 Resource	34
3.2.4 Soft-goal	35
3.3 Le Strategic Dependency Model	36
3.3.1 Caractéristiques du Strategic Dependency Model	36
3.3.2 Dependency link	36
3.3.3 Degré d'implication d'une dépendance	38

3.3.4	Exemple	40
3.4	Le strategic rationale model	41
3.4.1	Task decomposition link	41
3.4.2	Means-ends link	42
3.4.3	Contribute to Softgoal link	44
3.4.4	Exemple	47
3.5	Que peut apporter le modèle i^* aux théories des organisations? . . .	48
3.5.1	Les acteurs dans un organisation	49
3.5.2	Les buts	49
3.5.3	Le pouvoir	49
4	Les outils logiciels pour le modèle i^*	53
4.1	Redepend	53
4.1.1	Lancement de l'outil	54
4.1.2	l'interface	54
4.1.3	Utilisation	54
4.1.4	Enregistrement	58
4.1.5	Avantages et inconvénients	58
4.2	OME 3	59
4.2.1	Lancement de l'outil	59
4.2.2	Création d'un projet	60
4.2.3	Chargement d'un ou plusieurs modèles existants	60
4.2.4	Fermeture d'un projet ouvert	61
4.2.5	Création d'un nouveau modèle	61
4.2.6	Afficher un modèle existant	62
4.2.7	Modéliser un schéma i^* avec OME	62
4.2.8	Avantages et inconvénients	63
5	Possibles améliorations du modèle i^*	65
5.1	The Analytic Hierarchy Process(AHP)	67
5.1.1	Présentation intuitive de la méthode AHP	68
5.1.2	Démarche à suivre	68
5.1.3	Que peut apporter AHP au modèle i^* ?	75
5.2	Les attributs et conditions dans le modèle i^*	77
5.2.1	Introduction	77
5.2.2	Syntaxe et sémantique	78
5.2.3	Représentation graphique	80
5.2.4	Conclusion	80
6	Comment enrichir le modèle i^* grâce aux théories des organisations?	81
6.1	Les acteurs	81
6.1.1	Acteur interne et acteur externe	81
6.2	Les buts	82

6.2.1	Buts spécifiques et but de l'organisation	82
6.2.2	Buts de système et buts de mission	85
6.2.3	Les conflits entre buts	85
6.3	Le pouvoir	87
6.3.1	Les relations avec l'environnement	87
Conclusion		91
Bibliographie		93
A Le code de Redepend		95
A.1	Fonctions auxiliaires	95
A.2	Le code AHP du template visio	103
A.3	Le code AHP de la feuille de calcul Excel	116
B REFSQ'2002 workshop		129

Table des figures

3.1	Représentation d'acteur dans le modèle i^*	33
3.2	Représentation d'un "goal" dans le modèle i^*	34
3.3	Représentation d'une "task" dans le modèle i^*	34
3.4	Représentation d'une "resource" dans le modèle i^*	35
3.5	Représentation d'un "soft-goal" dans le modèle i^*	35
3.6	Représentation d'un "Dependency Link" dans le modèle i^*	37
3.7	Représentation d'un "Goal-dependency" dans le modèle i^*	37
3.8	Représentation d'une "Task-dependency" dans le modèle i^*	37
3.9	Représentation d'une "Resource-dependency" dans le modèle i^*	38
3.10	Représentation d'un "Soft-goal-dependency" dans le modèle i^*	38
3.11	Représentation d'une "open dependency" dans le modèle i^*	39
3.12	Représentation d'une "committed dependency" dans le modèle i^*	39
3.13	Représentation d'une "committed dependency" dans le modèle i^*	39
3.14	Strategic Dependency Model	40
3.15	Représentation d'une "Task-decomposition" dans le modèle i^*	41
3.16	Représentation d'une "Means-end" dans le modèle i^*	42
3.17	Principaux "Means-end" dans le modèle i^*	43
3.18	Différence entre un lien "means-end" et un lien "task decomposition"	44
3.19	Représentation d'un "Softgoal-Task link" dans le modèle i^*	45
3.20	Représentation d'un "Softgoal-Resource link" dans le modèle i^*	45
3.21	Représentation d'un "Softgoal-Goal link" dans le modèle i^*	46
3.22	Représentation d'un "Softgoal-Softgoal link" dans le modèle i^*	46
3.23	Strategic Rationale Model	47
3.24	L'expertise dans le modèle i^*	50
3.25	Le contrôle et l'usage de l'information dans le modèle i^*	51
4.1	la fenêtre d'accueil de Redepend	54
4.2	l'interface de Redepend	55
4.3	Ajout d'un "goal"	56
4.4	Attacher un lien à une figure	56
4.5	Message d'erreur lors d'une liaison incorrecte	57
4.6	Ajustement du lien "contribute to softgoal"	57
4.7	Fenêtre principale d'OME3	59
4.8	Création d'un projet	60
4.9	Chargement d'un projet	60

4.10	Création d'un nouveau modèle	61
4.11	Afficher un modèle existant	62
4.12	Modéliser un schéma i*	63
5.1	Airport check-in	65
5.2	Select site for corporate relocation	70
5.3	Airport check-in	75
5.4	Objectif d'un médecin	76
5.5	Check-in system sans attributs et conditions	77
5.6	Check-in system avec attributs et conditions	77
6.1	Acteur interne et acteur externe	82
6.2	Acteur et organisation	82
6.3	Strategic dependency model	83
6.4	Strategic rationale model	84
6.5	Buts de système et buts de mission	85
6.6	Contribute to soft-goal link	86
6.7	Acteur interne, externe et environnement	87
6.8	Fournisseur externe et gérant des stocks	88
6.9	Acteur analysant le marché	89

Introduction

Les ordinateurs deviennent de plus en plus présent dans la vie courante et les tâches exécutées par ceux-ci augmentent, s'imposent et entrent en relation avec la plupart des processus dans lesquels sont engagés les hommes. Il est important de tenir compte de ce facteur humain lors de la conception de système. Les relations qu'exercent les humains dans un système d'information sont analysés dans les théories des organisation. Les théories des organisations et les technologies d'information sont des disciplines qui permettent d'étudier l'aspect social et technique d'une organisation. Une collaboration étroite apparaît alors comme évidente entre ces deux domaines. Les systèmes d'information de dernière génération (e-business, knowledge management, customer relationship management, groupware, peer-to-peer, multi-agent) doivent correspondre d'une façon plus adéquate à leur environnement organisationnel afin de pouvoir s'adapter aux changements de celui-ci : marchés, conjoncture économique, restructuration, délocalisation... En effet un problème traditionnel des méthodes de développement d'un système d'information est la discordance sémantique séparant l'analyse de la conception. Cette discordance peut mener à des incohérences cruciales par après. L'analyse identifie et modélise le contexte organisationnel d'un système tandis que la conception est traditionnellement pilotée par les techniques de programmation. Pour comprendre pourquoi les technologies de l'information et les théories des organisations devrait collaborer, il est important de souligner deux prétentions erronées dès le départ ¹. La première est que les technologies de l'information et les théories des organisations couvrent deux champs distincts. La deuxième est que ces deux disciplines avaient auparavant de faible relations et que maintenant elles doivent apprendre de l'un et de l'autre de manière égale.

Bien qu'il y ait des différences entre les technologies de l'information et les théories des organisations, la frontière entre ces deux domaines n'a pas été longtemps brouillée. Ces deux disciplines sont difficiles à séparer parce que certains chercheurs travaillant dans le domaine des technologies d'information ont eu au départ une formation en organisation et d'autres ont été fortement influencés par la littérature sur les théories des organisations. De plus maintenant avec la complexité des systèmes d'informations, les ingénieurs seraient incapables de remodeler et informatiser tout un système de manière efficace sans avoir une compréhension générale

¹[Orl01]

des principes qui gouvernent les composants, les acteurs et les interactions entre eux dans un système. Cependant dans les méthodes d'analyse classique cette notion d'acteur, accomplissant des objectifs et interagissant avec les autres acteurs du système, pourtant primordial, fait souvent défaut. L'analyse des exigences ne tient pas suffisamment compte de ce concept issu des théories des organisation

L'analyse des exigences est une étape primordiale dans le développement d'un système d'information. La réussite d'un projet dépend en grande partie de celle-ci. Prenons l'exemple dans le domaine médical, le système mis en oeuvre dans les hopitaux à savoir le traitement des patients, c'est-à-dire les processus allant de l'opération au remboursement de la mutuelle sont un mélange de processus informatiques et humains. Quand les ordinateurs sont introduits dans un système d'information pour automatiser certaines tâches accomplies par les êtres humains c'est généralement avec une intention d'améliorer le processus en entier avec des critères de vitesse de précision, de fiabilité, etc... C'est pourquoi, pour répondre à ces critères, il faut bien cibler dès le départ les exigences d'un système. Le rôle de l'analyse des exigences ("Requirements Engineering") y est donc primordial. Malheureusement, les méthodologies d'analyse et conception existantes prennent mal en compte le traitement de l'évolution dynamique des systèmes. Un autre problème peut être souligné. En effet, les techniques d'analyse, plus particulièrement celles d'ingénierie du cahier de charge du système à construire, reconnaissent l'importance de modéliser le contexte organisationnel par des concepts sociaux et intentionnels. Le modèle *i** justement permet de réaliser cela aisément.

De plus, quand on essaie de dessiner un processus, il y a généralement plusieurs alternatives avec une implication différente pour chaque partie qui fait partie du processus. Les systèmes d'information ainsi développés jouent un rôle vital dans les organisations. Le succès des systèmes informatiques dépendent d'un grand nombre de facteurs qui vont du système technique à l'environnement organisationnel humain. Les agents d'une organisation peuvent avoir un certain nombre de préoccupations qui peuvent altérer l'environnement de travail. Par exemple avec l'introduction d'un nouveau système d'information il peut y avoir des changements dans les habitudes de travail. Des tâches peuvent être enrichies d'autres réduites. Il peut aussi y avoir du changement dans les relations de pouvoir. La liberté individuelle de chacun peut aussi être réduite. Les modèles conventionnels utilisés dans le développement de système d'information ont été développés à la base pour répondre à des besoins purement techniques et ne fournissent pas généralement une description détaillée de l'organisation d'un point de vue social. Ils ne tiennent généralement pas compte des impacts que l'organisation pourrait avoir sur un système. Un modèle qui tiendrait compte de ces aspects organisationnels permettrait une plus grande préoccupation de ces aspects lors du développement du système et faciliterait l'introduction du nouveau système d'information dans celui déjà existant. Les techniques de modélisations classiques ont besoin d'être étendues afin de rendre compte de l'organisation

existante et notamment des relations qu'exercent les acteurs d'un système. C'est dans cette optique que le modèle i^* a été développé. Le modèle i^* est fondé sur la notion d'acteurs ("actors") et de dépendance entre ceux-ci en terme de but ("goal"). Il permet de mettre en évidence les dépendances entre les acteurs d'un système.

Chapitre 1

Le modèle i^*

1.1 Motivation

Le modèle i^* a été développé à l'Université de Toronto par Eric Siu-Kwong Yu, John Mylopoulos et Chung¹. Il a été créé pour supporter la modélisation des organisations et aider à comprendre le raisonnement dans les relations qu'entretiennent les acteurs stratégiques dans un système. Le modèle i^* , comme je l'ai mentionné dans l'introduction, adopte une perspective orientée sur la notion d'*acteurs* ("actors") qui dépendent l'un de l'autre pour des buts à accomplir, des tâches à exécuter, des ressources à fournir. Le nom i^* prononcé "i-star" dérive en anglais de "distributed intentional" c-à-d que les systèmes d'information impliquent des acteurs qui ont une liberté d'actions mais dépendent les uns des autres en terme de but à accomplir, de tâches à exécuter et de ressources à fournir. Le modèle i^* est donc particulièrement pertinent par le fait qu'il met en évidence le concept d'acteur avec les buts que celui-ci recherche et les relations de celui-ci avec les autres au sein d'un système. Ce modèle permet de mettre en évidence les conflits potentiels de ces relations et de ces buts. Le modèle i^* peut donc faire le lien avec les théories des organisations où la notion d'acteur est primordial. Ce modèle est donc un premier pas pour une collaboration plus efficace entre les sociologues et les informaticiens.

Le modèle i^* inclu, d'une part, un "Strategic dependency model" pour décrire le réseau de relations existant parmi les acteurs d'un système, et d'autre part un "strategic rationale model" pour le raisonnement de chaque acteur au sein d'un système. Ces modèles peuvent couvrir les domaines d'application incluant "requirements engineering", "business process reengineering", "software process" et l'analyse des organisations. L'objectif de i^* est de fournir un modèle conceptuel riche pour modéliser un système qui implique de nombreux participants (à la fois des ordinateurs et des humains). Ceci permet l'identification des caractéristiques de modélisation pour décrire les processus et les raisonnements qui sont derrière ceux-ci et fournir une représentation formelle du système et sa sémantique. Le modèle i^* per-

¹[Yu95]

met une profonde compréhension du "pourquoi" qui ne peut pas être exprimé dans dans un modèle conventionnel qui accepte seulement la description d'entité et d'activité dans une organisation et ses interrelations(notamment UML) . Le modèle i^* permet de reconnaître les motivations, les intentions et le raisonnement des acteurs sous la surface des caractéristiques d'un processus. Le modèle est composé de deux sous-modèles à savoir le "strategic dependency model" et le rationale dependency model".

1.2 The Strategic Dependency Model

Le "Strategic Dependency Model" concrétise un réseau de relations de dépendances parmi les acteurs d'un système. La signification intuitive de dépendance est qu'une personne dépendant de quelqu'un d'autre ou de quelque chose(appelée respectivement "depend", "dependee" et "dependum") peut accomplir certains buts ou objectifs qu'elle serait incapable d'accomplir seul. Si le "dependum" n'est pas fourni par le "dependee", la tâche que "depend" doit accomplir, pourrait échouer ou être compromise.

Le "Strategic Dependency Model" a pour but de capturer la préoccupation principale de chaque acteur d'un système en laissant de côté les détails non essentiels.

Le "Strategic Dependency Model" montre qui dépend de "qui" ou de "quoi" "directement" ou "indirectement". Le modèle peut être utilisé pour explorer les opportunités qui sont ouvertes pour chaque acteur en faisant correspondre les "dependums" que les "dependees" et ceux que les "dependers" demandent. Le modèle peut être utile pour d'analyser la vulnérabilité d'un système en traçant des "chaines" de dépendances.

1.3 The Strategic Rationale Model

Dans le "Strategic Rationale Model" la tâche de chaque acteur peut être décrite explicitement en terme de processus et relations entre ceux-ci. Les principaux types de relations sont représentées par des liens "means-ends" et des liens de "task-decomposition". Les liens "Means-end" peuvent être vus comme une application de règles génériques dans un contexte particulier. Les processus cités plus haut incluent quant à eux des sous-buts (subgoals), des sous-tâches (subtasks) et des buts dont au moins un des aspects ne peut pas être décrit de manière concrète(softgoal). Chacun de ces éléments est inclu seulement s'ils est considéré comme suffisamment important dans l'accomplissement du but principal. Les agents d'un système peuvent accomplir quelque chose par eux-mêmes ou en dépendant d'autres agents. C'est pourquoi le "Strategic rationale model" reprend telle quelle la notion de dépendance du modèle précédent. Une collection de processus interconnectés entre eux servant pour les

besoins d'un agent est appelée une "routine". Un agent a souvent plus d'une routine pour accomplir quelque chose. La création d'un système d'information implique la modélisation de routines existantes (par ex. en demandant des questions du type "comment" et "pourquoi") et la découverte de nouvelles et meilleures routines.

Le "Strategic Rationale Model" offre donc selon Eric Yu 4 niveaux d'analyse². Au premier niveau on peut dire qu'un acteur a la *capacité (ability)* d'accomplir quelque chose s'il a une procédure définie pour cela. Ensuite au second niveau on peut vérifier si la procédure est *faisable (workable)* c-à-d si elle est réductible à des processus faisables à travers des liens "means-ends", "task decomposition" ou des liens de dépendance. Au troisième niveau on peut vérifier si la procédure est *viable (viable)* avec le respect de répondre à un critère qualitatif. Finalement au dernier niveau, on peut vérifier si les suppositions impliquées dans le raisonnement au sujet de la procédure est *crédible (believable)*, c-à-d suffisamment justifiée.

Ce modèle a donc pour but de fournir un support interactif pour un style de raisonnement "argumentatif" et non un raisonnement totalement automatique. Ce modèle est supposé être largement basé sur des jugements, des observations et de fréquentes connaissances incomplètes du système. Le but de ce modèle est de fournir un modèle formel qui peut mener à un support efficace pour aider les humains à manipuler, organiser, contrôler et tirer des conclusions de cette connaissance.

1.4 Travaux accomplis avec le modèle i^*

Depuis la thèse de doctorat de Eric Yu en 1995 le modèle i^* a fait l'objet de nombreuses études. Le modèle i^* est le modèle principal lors de l'utilisation d'une approche orientée agent, centrée sur les aspects intentionnels des agents. Il fait aussi partie comme extension de plusieurs modèles "Goal-Oriented Requirements Engineering (GORE)", notamment le modèle *NFR* et l'approche *KAOS*. Plusieurs thèses de doctorat ont été faites en utilisant ce modèle. Il s'agit de :

- Carolyn B. Seaman (1996) Organizational Issues in Software Development: An Empirical Study of³. CS-TR-3726. University of Maryland, USA
- Gary Katzenstein (1997), Carnegie Mellon University, USA. Beneath the Surface of Organizational Processes: A Social Context Framework for Business Process Redesign. ACM Transactions on Information Systems, 18(4), 383-422.
- Michaël Petit (1999) A Multi-Formalism and Component-Based Approach to the Formal Modeling of Manufacturing Systems Requirements⁴. University of Namur, Belgium.

²[Yu95]

³ftp://ftp.cs.umd.edu/pub/sel/papers/seaman_tr.ps.Z

⁴<ftp://ftp.info.fundp.ac.be/pub/users/mpe/thesis-petit.ps.gz>

- Stefanie Kethers (2000) Multi-Perspective Modeling and Analysis of Cooperation Processes⁵. Technical University of Aachen (RWTH), Germany.

Le modèle i^* est aussi la base et le point de départ du projet *TROPOS*⁶ qui développe une méthode pour les systèmes orientés agents. Les participants à ce projet sont :

- Tropos at the University of Toronto, Canada⁷
- Tropos at the University of Trento, Italy⁸
- Tropos at the Technical University of Aachen (RWTH), Germany⁹
- Tropos at the Federal University of Pernambuco (UPFE), Brazil¹⁰

Le modèle i^* est utilisé dans de nombreux projets. Citons par exemple quelques projets réalisés par le "Human-computer interaction design" de la "City University" à Londres à savoir d'une part le projet "eurocontrol", projet à l'échelle européenne visant la gestion du trafic aérien en Europe et notamment CORA-2 (Conflict Resolution Assistant) visant à éviter les collisions entre deux appareils¹¹ et, d'autre part le projet "SIMP project" (Systems Integration for Major Projects)¹² visant la gestion des manoeuvres de sous-marins.

1.5 Autres recherches déjà accomplies dans ce domaine

Des travaux similaires dans le domaine de l'analyse des exigences ont déjà été accomplis dont notamment la méthode *KAOS*¹³ développée par Axel Van de l'Université Catholique De Louvain et de la méthode *NFR* développée par Lawrence Chung¹⁴ de l'Université du Texas et Eric Yu. Ces méthodes contrairement au modèle i^* ne sont pas des modèles dont l'acteur est le centre de l'analyse. Pour la méthode

⁵<http://www.bth.rwth-aachen.de/job/fastnew.pl?start=1&access=0&base=Elektronische+Dissertationen&job=Kurzanzeige&index=11&tz=1&showmenue=0&sb=edit+00000130>

⁶<http://www.cs.toronto.edu/km/tropos/>

⁷<http://www.cs.toronto.edu/km/tropos/>

⁸<http://www.science.unitn.it/~pgiorgio/aose/>

⁹<http://www-i5.informatik.rwth-aachen.de/lehrstuhl/projects/tropos/>

¹⁰<http://www.di.ufpe.br/~jbc/>

¹¹De plus amples informations à propos de ce projet peuvent être trouvées ici : <http://www-hcid.soi.city.ac.uk/CORA-2.htm>

¹²Une explication très détaillée de ce projet peut être trouvée ici : <http://www.dcs.qmul.ac.uk/~simp/>

¹³[Lam]

¹⁴[CNM99]

KAOS, il s'agit en fait d'une méthode décrivant les concepts de but, de mission qu'un acteur doit accomplir. La méthode *NFR* quant à elle met en évidence aussi la notion de but, mais des buts qui peuvent être définis formellement. La méthode *NFR* est présentée ici parce que le modèle i^* en a hérité certains concepts.

1.5.1 La méthode *KAOS*

La méthode *KAOS* fournit un langage de spécifications, développé pour représenter les questions "qui", "pourquoi" et "quand" en plus du classique "quoi" dans les méthodes de spécification classique. Le langage fournit donc un concept riche pour capturer les exigences, les spécifications en terme de buts, actions, acteurs, contraintes, etc...

Un modèle *KAOS* est un réseau sémantique dans lequel les noeuds sont des concepts (c-à-d des objectifs à atteindre, les contraintes à respecter, les actions à accomplir, etc...) et les connexions entre noeuds des associations entre les concepts. Le cœur de la méthode *KAOS* consiste en l'application d'une grille de lecture particulière lors de la modélisation. La grille de lecture *Kaos* invite l'analyste à définir un réseau de concepts. La grille de lecture prédéfinit différents types de concepts tels que les buts, les agents, les entités, les actions, les associations, les scénarios. La grille de lecture précise également comment relier les concepts identifiés au travers de relations dont le type est aussi prédéfini. Parmi celles-ci, mentionnons la relation de raffinement de buts en sous-buts, la relation de conflit entre buts, la relation d'opérationnalisation entre un but d'une part et d'autre part un ensemble d'objets (entités, agents) et d'actions pour garantir la satisfaction du but, etc. Chaque concept identifié est décrit par un ensemble de caractéristiques prédéfinies (nom du concept, catégorisation, définition informelle, etc.) et par un ensemble de caractéristiques définies par l'analyste. Chaque concept pourra de plus faire l'objet d'une définition formelle.

Les buts pourront être formalisés mathématiquement (logique temporelle du premier ordre avec contraintes temps réel. Les actions seront spécifiées au moyen d'un contrat qui précise les conditions d'activation (préconditions) et les résultats de cette activation (postconditions); les objets seront spécifiés en décrivant des propriétés logiques qui s'y appliquent toujours (invariants). Dans ces deux derniers cas, le langage permet de distinguer les clauses qui relèvent de la théorie du domaine de celles qui sont rajoutées pour satisfaire un objectif.

La méthode *KAOS* propose également des guides méthodologiques pour orienter l'ordre dans lequel une grille de lecture peut être complétée; différents ordres sont possibles et ont été étudiés dont un ordre basé sur l'identification des buts et un autre guidé par les scénarios. La méthode *KAOS* a fait l'objet de nombreuses recherches pour laquelle la communauté scientifique a témoigné d'un grand intérêt à

en juger par le nombre de publications qu'elle a suscitées. Une méthode, aussi élégante et puissante soit-elle, ne connaîtra de véritable succès que si un grand nombre de professionnels sont convaincus de la possibilité de l'utiliser sur des cas industriels. C'est dans le but de préparer ce développement que l'Université catholique de Louvain, où est effectuée la recherche sur la méthode, a délégué à son centre de diffusion des technologies de l'information, le CEDITI-UCL¹⁵, la mission de réaliser un outil supportant la méthode Kaos et d'intéresser des partenaires industriels à utiliser la méthode. Ce projet de transfert a été financé par les fonds européens Objectif I et la Région wallonne sur la période 1995-1999. Il a conduit à des résultats probants.

1.5.2 La méthode NFR

La méthode *NFR* est utile pour le développement de solutions qui exigent des critères de qualités, des contraintes comme la performance, la sécurité... Ces exigences s'appellent les *spécifications non fonctionnelles* ou, en anglais, *Non Functional Requirements*. Les spécifications non fonctionnelles sont donc des spécifications qui décrivent non pas ce qu'un logiciel ou système doit faire (spécifications fonctionnelles) mais comment il pourrait le faire. Ces spécifications sont cependant difficiles à tester, c'est pourquoi elles sont souvent évaluées subjectivement. Les spécifications non fonctionnelles constituent les buts dans *NFR*.

La méthode *NFR* a pour but l'amélioration d'un système d'information existant en terme de spécifications non fonctionnelles. La méthode *NFR* consiste donc en la constitution d'un graphe représentant les interdépendances entre les buts non fonctionnels recherchés. Chaque but peut être décomposé en sous-buts plus spécifiques. Ces buts sont reliés par des liens. Ces liens peuvent représenter des contributions positives, négatives entre les buts, des conflits etc... Les buts non fonctionnels sont difficiles à définir rigoureusement. En effet, la qualité la performance sont des critères assez flous. C'est pourquoi la méthode *NFR* a défini un nouveau concept qui sera repris intégralement dans le modèle i^* . Ce concept est la notion de "softgoal". Les buts dans la méthode *NFR* s'appellent donc des "soft goal" en anglais car certains aspects n'ont pas pu être définis rigoureusement. Une explication plus détaillée de ce concept est expliquée au chapitre 3.

La méthode *NFR* ne prend pas en compte le concept d'acteur. Pour remédier à cela il a inclus entièrement le modèle i^* comme extension du modèle initial.

¹⁵www.cediti.be

Chapitre 2

Les théories des organisations

Une organisation est un groupe de personnes qui travaillent ensemble pour atteindre un but commun. Ou encore selon Mintzberg une organisation est *"la somme totale des moyens employés pour diviser le travail entre tâches distinctes pour ensuite assurer la coordination nécessaire entre ces tâches"*. Une organisation est donc représentée comme un système d'acteurs. Ce système est le produit émergent irréductiblement contingent des interactions que ses membres entretiennent les uns avec les autres ainsi qu'avec des acteurs placés dans son "environnement"¹. Dans ce chapitre je vais mettre en évidence les concept d'acteur, de pouvoir et de buts, concept que le modèle i^* prend directement en compte.

2.1 L'acteur et sa stratégie

Pour la théorie classique des organisations dans ses formes anglo-saxonnes (Taylor) et françaises (Fayol), le comportement humain ne constituait pas vraiment un problème. Avec le postulat de l'homo economicus, elle disposait en effet d'une théorie des motivations qui rendait les comportements humains parfaitement prévisibles, chaque agent étant censé être à tout moment rationnel, c'est - à - dire répondre de façon, stéréotypée aux changements des conditions techniques. C'est le mérite des célèbres expériences de la Western Electric que d'avoir permis la réfutation empirique de ce postulat, et du coup, d'avoir fait de la complexité des comportements humains dans les organisations un sujet de réflexion. Cette découverte correspondait à un enrichissement notable de la vision de l'homme au travail : l'individu n'est pas mû seulement par l'appétit du gain, il est motivé aussi par son affectivité et par ses besoins psychologiques plus ou moins conscients. Les travaux empiriques sur le travail et sur la bureaucratie issus du courant des relations humaines mettent en évidence la multiplicité des motivations des membres d'une organisation².

¹[CF77]

²[LM99]

L'organisation pose donc problème pour l'acteur car son fonctionnement repose sur une logique d'écart entre la théorie et la réalité, c'est-à-dire d'une part entre une rationalité surévaluée par l'admiration et la confiance des acteurs pour l'efficacité des résultats collectifs, et d'autre part la complexité des comportements humains, lesquels sont toujours contingents du besoin de liberté. Liberté de "battre le système, d'agir, de calculer, de s'adapter, contre tout moyen déterministe.

Pour l'analyse des comportements humains, il faut écarter les raisonnements a priori, qui considèrent seulement l'acteur et non la contingence des comportements dans un groupe, et réduisent ainsi les contraintes de l'organisation à un fait mécanique. Ainsi ,

- selon Chris Argyris qui a tenté de hiérarchiser les besoins psychologiques humains à partir du principe de "congruence", s'inspirant de la théorie motivationnelle de Maslow, l'acteur négocie seul avec l'organisation ;
- de même que le schéma de l'économie de marché - où rétribution équivaut à contribution - reste une théorie normative qui prévoit un cadre de référence.

Or, par définition, le groupe est un construit humain, dont la rationalité est proprement relative aux membres qui le constituent et à la différence de leurs comportements - s'échelonnant des actions stratégiques coordonnées à l'acteur apathique selon leurs opportunités et capacités, soit, en fonction de leur situation stratégique par rapport à la production, de leur degré de qualification professionnelle, et de leur degré d'intervention dans le groupe. Toute démarche stratégique devra relier la conduite de l'acteur au contexte, donc aux réductions organisationnelles. L'être humain est incapable d'optimiser comme le pensent March et Simon car sa rationalité reste limitée par sa marge de liberté et d'informations. Les acteurs sont des construits sociaux. La stratégie de l'acteur ne peut donc se concevoir seulement en termes d'objectifs clairs et de projets cohérents mais comme un jeu dans l'organisation, contingent au comportement et au vécu du participant. Ainsi, la stratégie de l'acteur revêt deux aspects : offensif pour saisir les opportunités et contraindre, et défensif pour agir et échapper aux contraintes. Cette idée compromet l'utilité d'un organigramme et pose la question de savoir qui domine en réalité.

Une typologie des acteurs peut être aussi réalisée. En effet, dans les description qu'il propose des différents acteurs, Mintzberg ³ est particulièrement attentif au rôle que peuvent jouer le cas échéant, ceux qui se situent en dehors de l'organisation. On ne s'étonnera pas que la distinction de base de l'auteur oppose les acteurs internes, c'est-à-dire les "employés salariés" à plein temps", aux acteurs externes, à savoir les "non-salariés".

³[Min86]

2.2 Le pouvoir comme fondement de l'action organisée

2.2.1 Définition

Du point de vue de l'acteur, le pouvoir, en tant qu'action de groupes ou d'individus sur d'autres groupes ou individus, s'entend en terme de relation : instrumentale quand les acteurs sont motivés par un but, non transitive car une hiérarchie reste indispensable pour obtenir des actions, et enfin réciproque mais déséquilibrée afin d'obtenir des forces de pouvoir. On peut donc en déduire que le pouvoir est un concept caméléon, difficile à définir. Le concept de pouvoir a deux face : la face "effet", c'est-à-dire ce que permet de faire le pouvoir, et la "cause", c'est-à-dire ce qui procure du pouvoir à un acteur ou un groupe d'acteurs.

- Côté effet, nous pouvons définir le pouvoir comme la capacité de certains acteurs ou groupe d'acteurs d'influencer ce que fait l'organisation et comment elle le fait dans un sens désiré" ou encore d'agir sur d'autres individus ou groupes. C'est pour le dire autrement, la capacité de faire coïncider ce que fait l'organisation ou ce que font les individus avec les intérêts ou les buts poursuivis par un groupe d'acteur ou un acteur particulier.
- Côté cause, le pouvoir réside implicitement dans la dépendance des autres. Il y a pouvoir quand un groupe d'acteurs ou un acteur est capable d'introduire une "brèche" ou une incertitude cruciale pour l'organisation et ses membres, qui rend ces derniers dépendants à l'égard de ceux qui la maîtrisent. C'est cette dépendance qui permet à certains groupes d'acteurs d'influencer l'organisation, d'y exercer du pouvoir. Le pouvoir met donc en avant le caractère *relationnel* de celui-ci que l'on peut résumer en soulignant que le pouvoir n'est pas un attribut d'une personne mais une relation.

Cette définition du pouvoir n'est pas en effet quelque chose dont certains sont pourvus d'autre non, quelque chose que l'on a définitivement ou pas. Au contraire, le pouvoir est quelque chose qui se construit progressivement dans les relations que nous entretenons avec les autres, en fonction de la dépendance que nous sommes capables d'introduire à notre égard. Ces relations de pouvoir sont généralement des relations de type conflictuel puisque, de leur côté, les autres tentent également, d'exercer du pouvoir à notre égard.

Il importe de bien réaliser que ces relations de pouvoir sont partout présentes dans l'organisation et ne sont pas des phénomènes exceptionnels. Ainsi, par exemple, un chef de projet ou un chef de service qui dépend, pour la réalisation des objectifs qui lui sont fixés par la direction, du bon vouloir de son personnel, de leur volonté à agir et à mettre tout en oeuvre pour la réussite de ces

objectifs, est inévitablement confronté à ces phénomènes de pouvoir dont il doit tenir compte dans la conduite intelligente de son personnel. Le pouvoir réside donc dans la capacité de rendre dépendant.

2.2.2 Les sources de pouvoir

Examinons à présent ce qui permet la dépendance et par voie de conséquence le pouvoir en organisation. La dépendance s'appuie, dans son fondement, sur la détention par un groupe d'acteurs ou un acteur d'un *ensemble de ressources dont dépendent les autres pour l'atteinte de leurs propres objectifs*.

Nous distinguons deux sources de pouvoir les ressources formelles, à savoir celles qu'un acteur peut tirer d'un droit ou d'une autorité conférée, et de l'autre côté, les ressources informelles, qu'un acteur peut construire en situation.

Parmi les ressources formelles, il y a essentiellement celles qui découlent de prérogatives légales, de droits exclusifs ou de l'apanage d'imposer des choix. Pour créer un dépendance, source de pouvoir dans une organisation, ces ressources formelles doivent être réelles, c'est-à-dire pouvoir effectivement servir dans la relation. Or ce n'est pas toujours le cas. Ainsi, dans de très nombreuses administrations publiques, le pouvoir hiérarchique du chef de service ou du directeur de département n'est pas un pouvoir effectif, dans la mesure où celui-ci n'a que très peu de moyens d'agir sur les promotions, les mutations, ou encore les licenciements de personnel, ces moyens d'action étant gérés de manière centralisée par les Secrétariats généraux et ce, pour éviter l'arbitraire des décisions et garantir l'équité de traitement de fonctionnaires. A travers ce bref exemple, on comprendra que l'autorité hiérarchique conférée par son statut à une personne ne peut se confondre avec le pouvoir, produit d'une relation effective de dépendance.

Les ressources formelles vont de pair avec un exercice du pouvoir qui s'ancre dans une certaine légitimité. Il en va tout autrement pour les ressources informelles. Celles-ci conduisent à l'exercice d'un pouvoir non légitimement reconnu (ou non-officiellement) mais pourtant bien réel dans ses effets sur l'organisation et les membres de celle-ci. Ce pouvoir se construit généralement des *zones d'incertitudes*⁴. Puisque ces sources de pouvoir correspondent à des possibilités d'action, les zones d'incertitude, contrôlées, deviennent pertinentes en ce qu'elles élargissent la marge de liberté des participants, leurs enjeux. En effet, ils pourront à la fois jouer de plusieurs relations de pouvoirs, ainsi diversifier leurs domaines d'investissement et se fixer un horizon dans le temps pour leurs stratégies. Du point de vue de l'organisation, si les contraintes qui la conditionnent sont contrôlées par les ensembles de pouvoir, ceux-ci sont également régularisés dans leur déroulement par la

⁴[CF77]

structure de l'organisation et dépendent de la volonté des acteurs de se mobiliser. Ainsi, l'organisation établit des canaux de communication entre les membres, et assoit son autorité légitime par un système de sanctions et de récompenses afin de développer l'esprit de compétition entre les acteurs. Afin de découvrir la façon dont l'organisation motive ses membres et infirmer Clegg pour qui il existe des inégalités structurelles au sein de l'organisation, on pourra donc établir une typologie des pouvoirs en fonction des différents types de sources d'incertitudes, à savoir :

- la maîtrise d'une compétence particulière,
- la relation de l'organisation à son environnement pour ses besoins en ressources ou pour vendre,
- la maîtrise de la communication et des informations,
- l'ensemble des règles de l'organisation pour supprimer ces incertitudes ou paradoxalement créer d'autres incertitudes dans la négociation de ces règles.

2.2.3 Conclusion

Toute organisation met ses membres en situation d'interdépendance déséquilibrée et ouverte, où chaque participant a besoin de la contribution de l'autre pour réussir. Il y a pouvoir parce que, simplement, les contraintes des uns ne sont pas celles des autres et que les impératifs et logiques des uns ne coïncident jamais totalement avec celles des autres. Comme relation, le pouvoir possède un lien irréductible avec l'(inter)dépendance, c'est-à-dire avec la coopération et l'échange. Même si cet échange est toujours et structurellement déséquilibré ; pas de pouvoir sans relation, pas de relation sans échange, pas d'échange sans négociation. C'est là la dimension instrumentale du pouvoir. On ne noue pas des relations de pouvoir gratuitement ou pour l'unique plaisir d'en avoir. On entre dans une relation de pouvoir parce qu'on doit obtenir la coopération d'autres personnes pour la réalisation d'un projet, quel qu'il soit. Le pouvoir peut et doit être défini comme la capacité d'un acteur à structurer des processus d'échange plus ou moins durables en sa faveur, en exploitant les contraintes et les opportunités de la situation pour imposer les termes de l'échange favorable à ses intérêts. C'est un échange négocié de comportements qui est structuré de telle sorte que les participants en retirent quelque chose, tout en permettant à quelqu'un (ou quelques-uns) d'en retirer davantage que les autres. L'autre source du pouvoir de chacun des participants est la liberté ou l'autonomie dont il dispose dans ses transactions avec les autres et qui détermine la prévisibilité de son comportement pour les autres. Le jeu de l'échange négocié de comportements peut donc bien être reconstruit autour d'une logique de l'imprévisibilité et de l'incertitude.

2.3 Les buts dans une organisation

2.3.1 Notion

Un but peut être défini comme tout ce qui motive l'action ou la décision en organisation. Ces buts sont par nature individuels ou collectifs. Collectifs quand ils sont partagés par un grand nombre d'individus. Les buts poursuivis peuvent se ranger en deux types : les buts de mission et les buts de système⁵.

Les buts liés à la mission décrivent la vocation externe de l'institution. Ils sont la raison d'être de l'organisation. Par exemple, le but de mission d'une école, c'est l'éducation des jeunes. Les buts déclarés sont le plus souvent des buts de mission. Ils sont généralement exprimés en termes vagues et imprécis. Ils sont donc sujets à diverses interprétations à l'intérieur et à l'extérieur de l'institution, ce qui peut être à l'origine de bien des conflits et des blocages. Il est assez difficile d'établir une typologie des buts de mission de l'organisation dans la mesure où ces buts sont très spécifiques à chaque organisation et à son objet social. Ces buts sont visibles dans les déclarations faites par ces organisations en regard à leur objet social.

Les buts liés au système sont ceux que l'institution poursuit pour assurer son existence et son développement. Ils concernent directement l'organisation et son personnel. Ces buts peuvent être la survie (but minimal), la croissance, le contrôle de son champ d'action, une utilisation efficiente des ressources. Ils existent dans toutes les organisations, mais ils ne sont pas toujours explicites ou conscients. Ils sont pourtant fondamentaux pour que l'organisation puisse poursuivre des buts de mission.

Les buts de système font l'objet d'une typologie plus fine⁶. On y distingue la survie, l'efficience, le contrôle de l'environnement et la croissance.

Les buts de survie

Les buts de survie sont les premiers buts poursuivis par toute organisation dans la mesure où la survie est la condition nécessaire à la poursuite de n'importe quel autre but organisationnel. Les buts de survie sont très présents et très mobilisants dans les entreprises en phase de démarrage. Dans certains cas, la poursuite de buts de survie peut avoir un effet sur la redéfinition des buts de mission.

Les buts d'efficience

Par but d'efficience, on entend l'amélioration du rapport entre les ressources utilisées par l'organisation et les résultats qu'elle obtient, ou sa performance. Les

⁵[Min86]

⁶[Min86]

buts d'efficience sont généralement très opérationnels et mesurables (rapport entre les coûts de production et le produit des ventes). Dans certaines organisations, les buts d'efficience peuvent être tellement dominants qu'ils écartent l'organisation de ses buts de mission.

Les buts de contrôle de l'environnement

Les organisations subissent généralement de très nombreuses pressions des milieux extérieurs. Un des buts de l'organisation est d'essayer de contrôler ces pressions, voire de les dominer afin de pouvoir rester maître de son fonctionnement et de ses finalités.

Les modalités de contrôle de l'environnement peuvent être très diverses et varier en fonction de la partie de l'environnement concernée.

- Vis à vis des organisations avec lesquelles l'organisation entretient des relations commerciales, le contrôle de l'environnement et notamment la maîtrise des incertitudes qui peuvent venir perturber les relations entre partenaires peut passer par des stratégies d'intégration verticale visant à incorporer en son sein des fournisseurs de biens ou services dont elle est dépendante; et/ou par des stratégies d'intégration horizontale et de fusion visant à absorber ou maîtriser les concurrents.
- Vis à vis des pouvoirs publics, les modalités de contrôle peuvent également être très diverses. Ainsi, la présence au sein du conseil d'administration d'un élément politique tampon est souvent utilisée comme moyen pour maîtriser l'incertitude provenant du monde politique. A contrario, l'entreprise peut également jouer la carte de sa représentation dans des instances politiques ou patronales dont elle dépend pour mener à bien son activité.

Les buts de croissances

Les buts de croissance se manifestent dans la volonté de l'entreprise d'augmenter ses ressources et/ou ses parts de marché et/ou son chiffre d'affaires. L'organisation recherche la croissance pour différentes raisons dont certaines sont évidentes et d'autres plus politiques. L'organisation recherche d'abord la croissance car celle-ci conditionne sa survie. Par ailleurs la croissance d'une entreprise est aussi un facteur qui permet de renforcer le pouvoir ou le contrôle qu'une entreprise peut exercer sur son environnement, qu'il soit commercial ou politique. Comme le souligne Mintzberg, *"Les grandes organisations ont plus de pouvoir et de prise sur leurs environnements. Elles sont davantage capables de résister aux pressions immédiates pour un changement et, de plus, disposent davantage de temps pour distinguer les menaces extérieures, et pour se préparer à les affronter"*⁷. Enfin la croissance est

⁷[Min86]

également poursuivie à des fins plus politiques, dans la mesure où elle peut signifier pour certains acteurs dans l'organisation une augmentation de leurs moyens et partant, un renforcement dans leur position politique dans l'organisation. Elle est également un moyen particulièrement efficace pour résoudre les tensions internes et les conflits politiques au sein des organisations.

2.3.2 Les rapports entre buts

Aux égards des différents buts organisationnels exposés ci-avant, il importe de se demander comment ceux-ci parviennent à coexister au sein d'une organisation. A ce niveau deux cas de figure peuvent caractériser cette coexistence⁸. Le premier est celui de la cohabitation pacifique entre les différents buts poursuivis par l'organisation, ceux-ci se renforçant mutuellement. C'est ce qu'on appelle un *système de buts intégrés*. Le second cas de figure caractérise la situation où des buts rentrent en conflit les uns par rapport aux autres, c'est le *système de buts conflictuels*⁹.

Les systèmes de buts intégrés

Les systèmes de buts intégrés peuvent se décliner de deux manières.

La première manière représente la situation où un seul but domine l'ensemble des autres et unit l'ensemble des membres de l'entreprise autour de cette orientation dominante. Ainsi dans certaines entreprises très marquées par la concurrence et dont la survie dépend de la maîtrise de celle-ci, l'ensemble de l'organisation peut apparaître comme entièrement mobilisée par cette bataille, celle-ci marquant aussi bien les discours officiels des dirigeants que les décisions prises quotidiennement par les différents départements.

La seconde manière, plus courante, est celle où différents buts coexistent, la poursuite des uns contribuant à la réalisation des autres. Ainsi, dans telle entreprise dont les buts de mission s'affirment très clairement comme étant ceux de mettre sur le marché des produits ou services marqués par le meilleur rapport qualité-prix, les buts de système poursuivis dans l'entreprise pourront venir appuyer cette recherche de l'excellence à travers des programmes de qualité totale(but d'efficience) et de contrôle sur les fournisseurs et les sous-traitants(but de contrôle de l'environnement), visant à diminuer l'incertitude de l'organisation, tant au niveau de la qualité que du prix du produit ou service offert.

Les systèmes de buts conflictuels

Ces situations de coexistence pacifique entre buts poursuivis dans l'organisation supposent l'existence, dans l'entreprise, d'un très large consensus entre acteurs au-

⁸[NP95]

⁹[Min86]

tour de ces buts. Toutefois, dans nombre d'entreprises, par simple fait que les acteurs dans l'organisation sont porteurs d'intérêts différents, voire divergents, les buts autour desquels se mobilise l'organisation peuvent apparaître moins consensuels. Cette situation est ce que Mintzberg appelle *les systèmes de buts conflictuels*.

Les conflits entre les buts poursuivis dans une organisation peuvent revêtir trois formes :

- Les conflits entre buts de mission.

De très nombreux exemples peuvent servir à illustrer ce premier cas de figure. Ainsi, par exemple, dans la plupart des universités, des conflits existent entre les deux buts de mission que sont d'une part l'enseignement et d'autre part la recherche. Ces deux missions fondent très légitimement l'activité de toute université; toutefois, avec la diminution des moyens alloués à l'enseignement universitaire, la plupart des universités visent à pallier ce déficit par un investissement plus important dans les activités de recherche, dans la mesure où celles-ci constituent une source de revenu additionnel pour ces institutions. Très concrètement, cette tendance provoque, dans les universités, des conflits internes importants entre membres, les uns mobilisés par la fonction de l'enseignement et critiquant le manque de professionnalisme et de qualité de l'enseignement prodigué par les autres, ces autres, plus mobilisés par la recherche, critiquant les premiers pour le manque de dynamisme dans la recherche et dans l'apport de moyens additionnels à l'université.

- Les conflits entre buts de système et buts de mission

Les conflits entre buts de systèmes et buts de mission peuvent également marquer profondément la paysage organisationnel. Ici encore, comme souligné dans l'exemple précédent, ces conflits sont généralement le fait de divergences entre acteurs porteurs de buts difficilement conciliables. Ainsi par exemple, dans un hôpital, les buts de mission poursuivis par le corps médical à savoir de prodiguer des soins médicaux de qualité et respectueux du malade peuvent-ils s'opposer aux buts de système poursuivis par les gestionnaires allant dans le sens d'une rationalisation des coûts de fonctionnement et un meilleur rendement des investissements.

- Les conflits entre buts de système.

Les conflits entre buts de système peuvent également perturber le paysage organisationnel. Ainsi, de très nombreuses entreprises sont marquées par des conflits entre ces deux buts de système que sont d'une part, l'efficacité de l'entreprise et, d'autre part sa croissance. La poursuite du but d'efficacité

suppose très souvent une certaine rationalisation des moyens tandis que le but de croissance nécessite, au contraire, des investissements porteurs d'avenir. Les conflits qui marquent aujourd'hui la plupart des sociétés européennes de chemin de fer ne font que traduire cette tension entre efficience accrue, motivée par des perspectives d'une éventuelle libéralisation du secteur et investissement nécessaire pour faire face à l'augmentation du trafic ferroviaire.

2.3.3 Buts spécifiques et buts organisationnels

Tout acteur en organisation, quel que soit son niveau hiérarchique, poursuit des buts de mission et des buts de système qui motivent son action.

Toutefois il existe des niveaux de buts très différents en organisation. Certains buts s'affirment comme étant des buts organisationnels parce que, d'une part, portés par les plus hautes instances de l'entreprise et d'autre part, par ce fait même, s'imposant comme un cadre stratégique de référence à la majorité de ses membres. Ces buts forment en quelque sorte le cadre stratégique dans lequel les différents acteurs de l'entreprise doivent se mouvoir. Comme nous l'avons vu ci-dessus, ce cadre stratégique de référence peut être harmonieux ou consensuel, il peut, à contrario, être traversé par des contradictions voire des oppositions qui marquent le fonctionnement quotidien de l'entreprise.

Au delà de ces buts organisationnels que l'on retrouve généralement dans les discours officiels, chaque acteur en organisation est également porteur d'un ensemble de buts de mission et/ou de système qui lui sont propres. La poursuite de ces buts, que nous avons qualifiés de spécifiques, motive l'acteur au quotidien. A nouveau les raisonnements que nous avons menés pour les buts organisationnels concernant les rapports entre buts(système intégré ou système conflictuel) sont valables également pour les buts spécifiques, les buts spécifiques des différents acteurs travaillant, par exemple, dans un même service pouvant être harmonieux ou, au contraire, conflictuels.

Qu'il s'agisse de buts organisationnels ou de buts spécifiques, la capacité de les faire reconnaître par d'autres en organisation et de les réaliser dépend du pouvoir qu'exerce l'acteur en organisation.

Chapitre 3

Les éléments du modèle i^*

Dans ce chapitre je vais décrire en détails tous les éléments qui font partie du modèle i^* . Ensuite je vais faire le rapprochement de ces éléments avec les théories des organisations de façon à permettre aux sociologues d'utiliser aisément ce modèle.

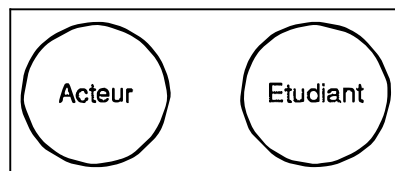
3.1 Acteurs

Un acteur est une entité active qui accomplit des actions pour réaliser ses objectifs en employant son savoir-faire. Les acteurs ont la liberté d'action mais dépendent l'un de l'autre pour accomplir un but donné ou effectuer une tâche. Un acteur n'accomplit pas uniquement des actions mais a des aspects "intentionnels" comme l'objectivité la rationalité et l'engagement. Ces aspects intentionnels associés avec chaque acteur peuvent être distingués en quatre éléments : "goal", "task", "soft-goal" et "resources". Les acteurs dépendent l'un de l'autre pour les buts à accomplir, les tâches à exécuter et les ressources à fournir.

3.1.1 Représentation

Un acteur stratégique dans le modèle i^* peut être représenté par un cercle contenant le nom de l'acteur.

FIG. 3.1 – Représentation d'acteur dans le modèle i^*



3.2 Éléments associés avec les acteurs

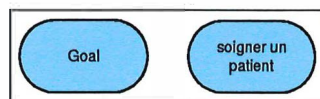
Les acteurs, dans le modèle i^* réalisent des objectifs. Ces objectifs peuvent être modéliser de 4 manières différentes.

3.2.1 Goal

Un "goal" représente condition ou un état qui peut être accompli ou non. Un "goal" ne décrit pas comment un état désiré ou une condition peut être accomplie mais décrit uniquement la condition ou l'état à atteindre. Ceci permet de dire qu'un "goal" peut être atteint de différentes manières.

Représentation

FIG. 3.2 – Représentation d'un "goal" dans le modèle i^*

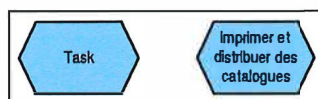


3.2.2 Task

Une "task" représente une manière particulière d'atteindre un objectif. Une tâche peut être en fait considérée comme une description détaillée de comment atteindre un objectif. Une tâche peut être aussi interprétée comme une activité qui change quelque chose dans le système.

Représentation

FIG. 3.3 – Représentation d'une "task" dans le modèle i^*

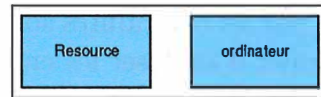


3.2.3 Resource

Les "resources" dans le modèle i^* sont utilisées pour modéliser les objets d'un système. Ces objets peuvent être physique ou informel. Les "resources" peuvent être considérées comme la disponibilité d'une certaine entité ou un produit fini d'une action, d'un processus ou d'une tâche.

Représentation

FIG. 3.4 – Représentation d'une "ressource" dans le modèle i^*

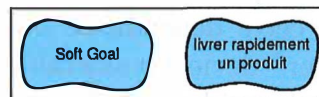


3.2.4 Soft-goal

Le concept de "softgoal" est issu de *NFR*. Les "softgoals" sont des "goal" dont une propriété ne peut être définie de façon précise. Pour illustrer ceci on peut prendre comme exemple le "soft-goal" "fournir de bon produit". Ceci est un "soft-goal" car le terme "bon" ne peut être défini de façon précise. On peut aussi utiliser un concept abstrait comme la "satisfaction", qui constitue aussi "un softgoal" car elle dépend d'une personne à l'autre. Ces concepts constituent comme on l'a vu précédemment les *spécifications non fonctionnelles*.

Représentation

FIG. 3.5 – Représentation d'un "soft-goal" dans le modèle i^*



3.3 Le Strategic Dependency Model

Le Strategic Dependency Model présente une description d'un processus en terme d'un réseau de relations de dépendance parmi les acteurs du système. Ce modèle a pour but de montrer que les acteurs peuvent agir librement dans les tâches qui leur sont confiées mais sont liés par les contraintes sociales (inter-acteurs). Grâce à ces concepts riches ce modèle permet d'analyser les enjeux, les opportunités et les faiblesses.

D'un point de vue organisationnel ce modèle permet d'analyser les buts individuels poursuivis par les acteurs du système ainsi que les relations de pouvoir que ceux-ci exercent entre eux

3.3.1 Caractéristiques du Strategic Dependency Model

Le strategic Dependency Model consiste en un jeu de noeud et de liens. Chaque noeud représente un "acteur" et chaque lien entre deux acteurs indique qu'un acteur dépend d'un autre par quelque chose qui lui est nécessaire pour qu'il atteigne son but recherché. On appelle l'acteur dépendant le "depender" et celui dont dépend celui-ci le "dependee". L'objet qui est au centre de cette dépendance est appelé le "dependum".

Une dépendance est intentionnelle si le "dependum" est d'une manière ou d'une autre liée à certains buts ou désir du "depender". En dépendant d'un autre acteur pour un "dependum", un acteur(le depender) est capable de poursuivre son but qu'il était incapable d'accomplir sans cette dépendance ou pas aussi simplement que cela. En même temps le "depender" devient vulnérable. En effet si le "dependee" n'arrive pas à fournir le "dependum", le "depender" en sera affecté dans sa capacité à accomplir son objectif. Dans le "Strategic Dependency Model", les objectifs internes du "depender" ne sont pas modélisés explicitement.

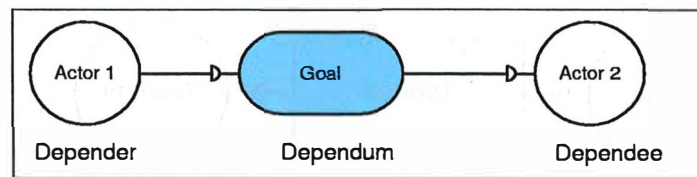
3.3.2 Dependency link

Le "Dependency link" est un des éléments constituant le "Strategic Dependency Model". c'est un lien entre deux acteurs qui indique qu'un acteur dépend d'un autre acteur par un "dependum" essentiel pour atteindre son objectif.

Représentation

Dans cette représentation, on peut y voir le "depender", le "dependee" et le "dependum"

FIG. 3.6 – Représentation d'un "Dependency Link" dans le modèle i*



On peut compter quatre "Dependency Link" basés sur le type du "dependum" :

Goal Dependency

Dans un "goal dependency" le "dependeur" dépend du "dependee" pour apporter un nouvel état dans le système. Le "dependee" a reçu la liberté de choisir comment le faire.

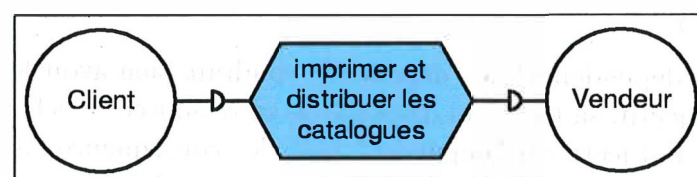
FIG. 3.7 – Représentation d'un "Goal-dependency" dans le modèle i*



Task Dependency

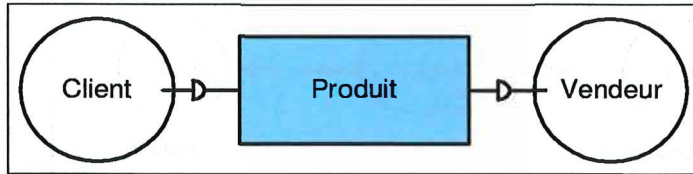
Dans le cas d'un "task dependency" le "dependeur" dépend du "dependee" pour exécuter une activité.

FIG. 3.8 – Représentation d'une "Task-dependency" dans le modèle i*



Resource Dependency

Dans un "resource dependency" le "dependeur" dépend du "dependee" pour la disponibilité d'un produit.

FIG. 3.9 – Représentation d'une "Ressource-dependency" dans le modèle i^* 

Soft-goal Dependency

Ici dans le cas d'un "softgoal dependency" le "depender" dépend du "dependee" pour un objectif dont une propriété ne peut pas être explicitement définie.

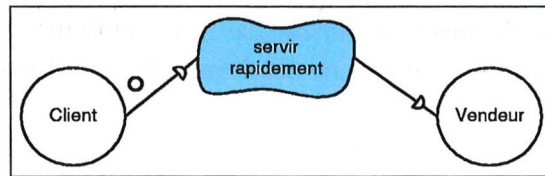
FIG. 3.10 – Représentation d'un "Soft-goal-dependency" dans le modèle i^* 

3.3.3 Degré d'implication d'une dépendance

Le modèle distingue aussi plusieurs degrés de dépendance. Du côté du "depender" une forte dépendance signifie que celui-ci est plus vulnérable et qu'il est préférable qu'il prenne de sérieuses mesures pour atténuer cette vulnérabilité. Du côté du "dependee", une forte dépendance implique que le "dependee" doit faire un grand effort pour essayer de délivrer le "dependum". Le modèle fournit trois degrés d'implication : "Open", "Committed" et "Critical".

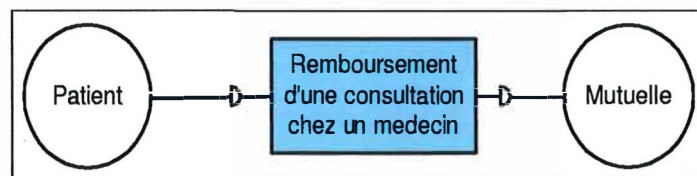
Open Dependency

Dans une "open dependency" le "depender" voudrait bien avoir le "dependum" pour réaliser son objectif, sa tâche ou disposer de sa ressource. L'échec quant à ceci va bien-sûr affecter l'objectif du "depender" mais les conséquences n'en seront pas dramatiques. Du côté du "dependee" l'"open dependency" est une affirmation que le "dependee" est prêt à accomplir le "dependum" pour le "depender".

FIG. 3.11 – Représentation d'une "open dependency" dans le modèle i^* 

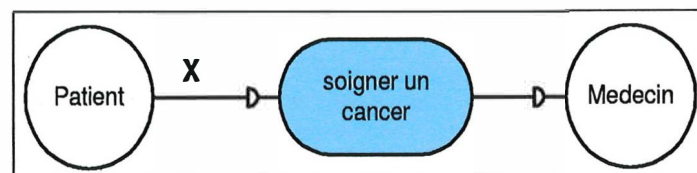
Committed Dependency

Dans une "committed dependency" le "dependeur" a des objectifs qui pourraient être affectés de manière significative si le "dependum" n'est pas accompli. Le "dependeur" doit s'investir considérablement dans sa tâche qui ne peut être abandonnée sans pertes pour le système. Du côté du "dependee" une "committed dependency" signifie que le "dependee" va essayer de faire de son mieux pour fournir le "dependum".

FIG. 3.12 – Représentation d'une "committed dependency" dans le modèle i^* 

Critical Dependency

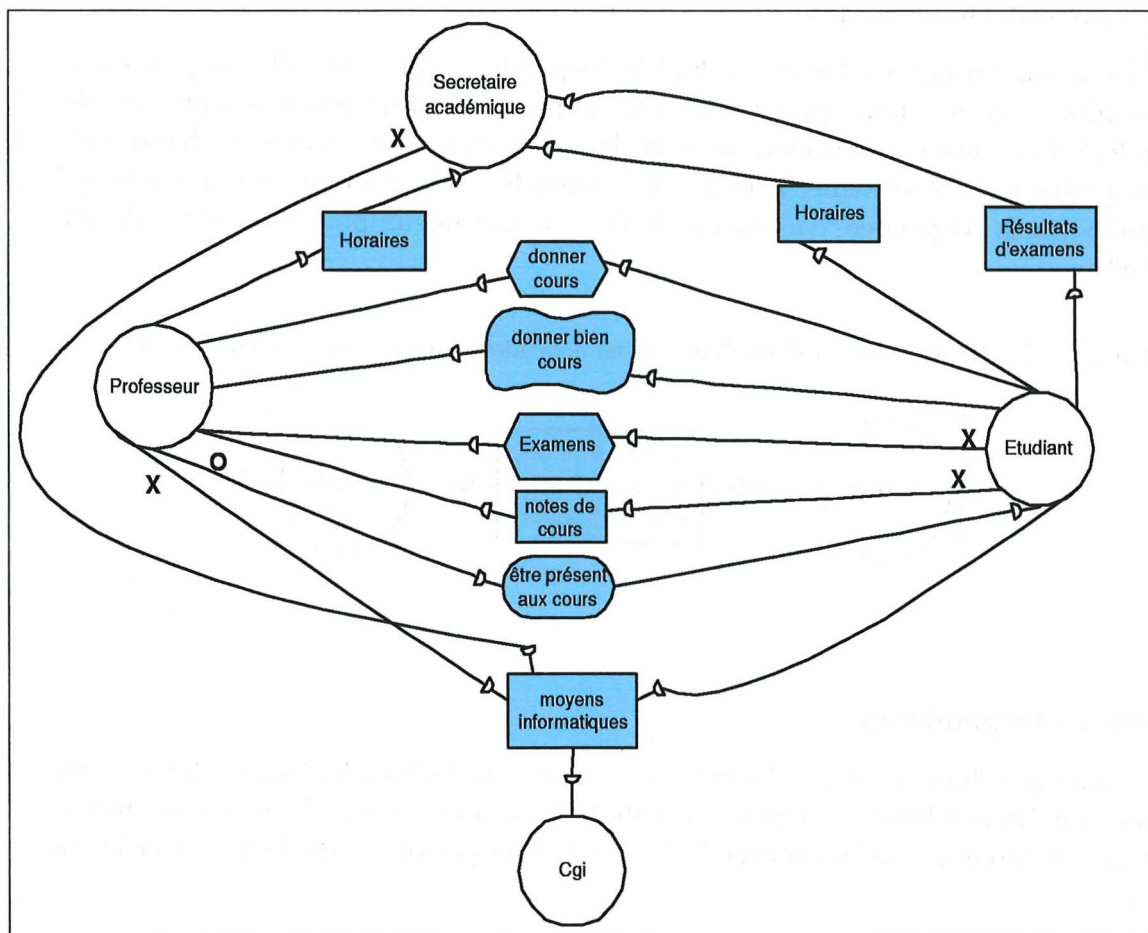
Dans une "critical dependency", la situation du "dépendeur" peut être très critique si le "dependum" n'est pas accompli. Cela peut engendrer de sérieux problèmes à tous les acteurs en relation avec le "dependeur" et parfois même bouleverser le système.

FIG. 3.13 – Représentation d'une "critical dependency" dans le modèle i^* 

3.3.4 Exemple

Voici le "strategic dependency model" pour l'institut d'informatique. L'organisation a été très simplifiée de façon à permettre une meilleure compréhension du modèle. Il se résume à quatre acteurs : Les étudiants, la secrétaire académique, les professeurs et le CGI.

FIG. 3.14 – Strategic Dependency Model



3.4 Le strategic rationale model

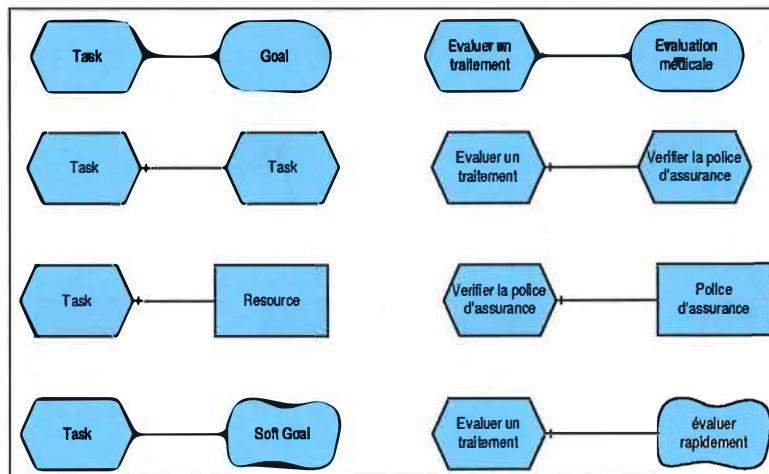
Le strategic rationale model fournit une description intentionnelle de processus en terme d'éléments du processus et de raisonnement à l'intérieur de ceux-ci. Alors que le "strategic dependency model" tenait un certain niveau d'abstraction en décrivant uniquement les relations externes entre les acteurs, le "strategic rationale model" va plus loin en permettant une plus profonde compréhension du raisonnement des acteurs stratégiques en terme de processus qu'ils accomplissent. Le "strategic rationale model" décrit les relations intentionnelles qui sont "internes" à chaque acteur. Le "strategic rationale model" considère uniquement ce qui est important pour influencer l'accomplissement d'un but. Le "strategic rationale model" inclut le "strategic dependency model" qui permet de voir qu'un acteur est capable d'accomplir quelque chose par lui-même ou en dépendant d'autres acteurs.

Les quatre types de noeuds vus précédemment à savoir les "goals", "softgoals", "task" et "resources" peuvent être liés par quatre type de relations : "dependency link" (identique au "strategic dependency link"), "task decomposition link", "means end link", "contribute to softgoal link".

3.4.1 Task decomposition link

Une tâche est modélisée en terme de ses décompositions en sous-composants. Ces composants peuvent être des "goals", "tasks", "resources" et/ou "softgoals". La relation entre ces composants peut être interprétée comme un *ET logique*, c-à-d que chaque sous composant constitue une partie de l'exécution de la tâche principale. Il peut y avoir quatre types de décompositions comme le montre le schéma ci-dessous :

FIG. 3.15 – Représentation d'une "Task-decomposition" dans le modèle i*



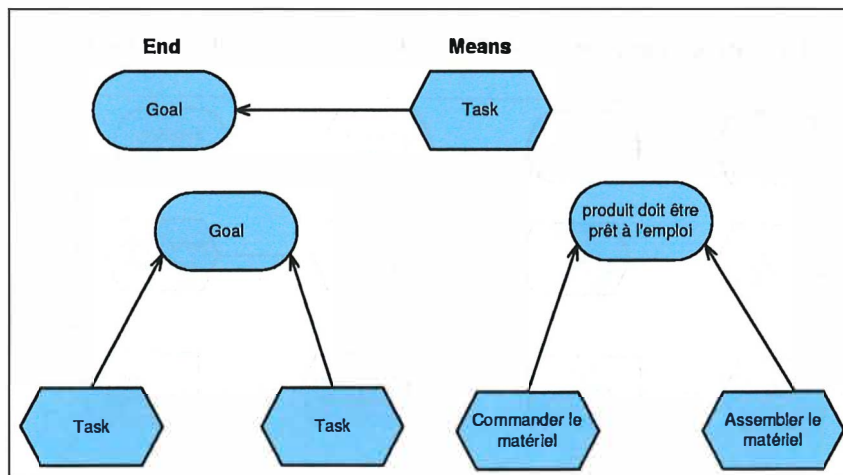
Ceci implique plusieurs choses :

- Lorsque un "goal" apparaît dans la décomposition, cela signifie que la façon dont le but doit être accompli n'est pas défini et donc différentes alternatives peuvent être prises en compte.
- Lorsque une "task" est introduite comme sous-tâche à une "task" principale cela signifie que la "task" principale est restreinte à une voie particulière quant à son exécution.
- Une "resource" dans la décomposition n'est pas considérée comme problématique en ce sens qu'on doit juste se poser la question quant à sa disponibilité.
- Quand un "softgoal" apparaît dans la décomposition il agit en tant que but qualitatif pour une tâche, et aussi comme un guide pour limiter le choix parmi les alternatives.

3.4.2 Means-ends link

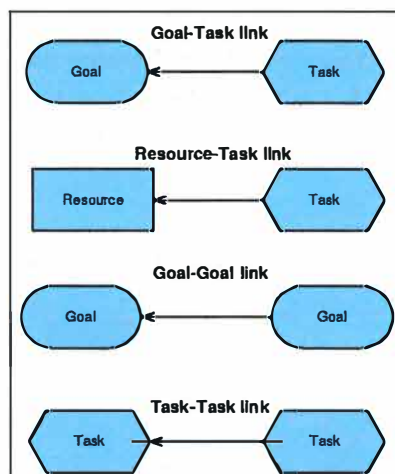
Le "strategic rationale model" fournit aussi différents "means-end link". Le "end" peut être un "goal", "task", "resource" ou "softgoal". Le "means" (moyen) est habituellement une "task", mais peut être aussi un "goal" ou "resource". Les liens "means-end" entre les différents "means" peuvent être considérés comme un *OU logique*.

FIG. 3.16 – Représentation d'une "Means-end" dans le modèle i^*



Les principaux liens "means-end" sont listés ci-dessous :

FIG. 3.17 – Principaux "Means-end" dans le modèle i*



Goal-Task link

Dans un "goal task link", le lien spécifie une voie pour atteindre l'objectif. Un goal peut avoir différents liens "means-end" représentant différents moyens pour atteindre l'objectif, comme on a vu précédemment. Dans le un "goal-task link", la "task" spécifie *comment* atteindre un objectif à travers sa décomposition en sous-composants.

Resource-Task Link

Dans le "resource-task link", le "end" est spécifié comme une "resource" et le "means" est décrit comme une "task". La "task" indique comment la ressource peut être obtenue. De nouveau la "resource" peut être atteinte par différents moyens, ainsi il peut y avoir plusieurs "means-end" pour la même "resource".

Goal-Goal Link

Le "goal-goal" link" permet la réduction de "goals" en "subgoals". Avec un "goal" au niveau du "end", le "means" peut être décrit comme la conjonction de "subgoals". Cela requiert qu'aucun autre élément("task", "resource", "sofgoal") doit être lié au "goal".

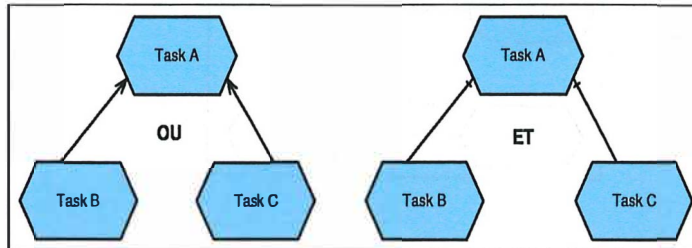
Task-Task Link

Si le lien a une "task" à chaque extrémité cela représente une méthode pour décomposer une "task" en de possible "subtask" alternatives, avec la relation logique OU. Bien qu'une "task" spécifie *comment* quelque chose doit être fait, il peut y avoir

différentes alternatives pour exécuter une "task", ainsi on peut rencontrer des liens "means-end".

Cependant il est important de bien voir la différence entre un lien "means-end" et un lien de "task decomposition". Le "task decomposition" implique une relation *ET* entre les "subtasks" tandis que un "means-end" montre un *OU logique* entre les "subtasks".

FIG. 3.18 – Différence entre un lien "means-end" et un lien "task decomposition"



Sur le schéma de gauche, la "task" A peut être exécutée en accomplissant la "task" B *ou* C tandis que dans celui de droite, la "task" A ne peut être accomplie que en exécutant la "task" B *et* C.

3.4.3 Contribute to Softgoal link

Un lien "Contribute to softgoal" est un "means-end" avec un softgoal lié au "end". Comme l'accomplissement d'un "softgoal" ne peut clairement être défini, un "means-end" pourrait porter à confusion. Ceci est dû au fait qu'aucun moyen ne peut être spécifié pour l'accomplissement d'un objectif qui ne peut pas être clairement défini.

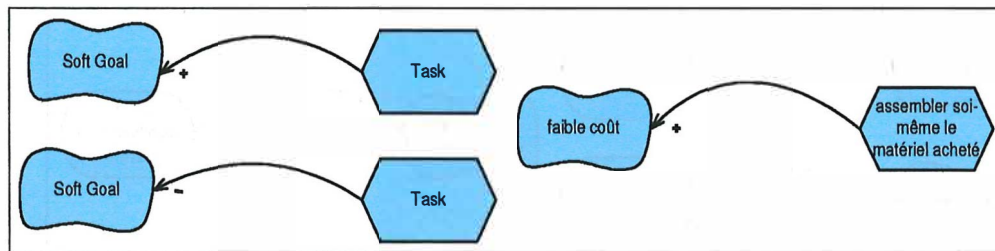
C'est pourquoi un nouveau lien a été créé et c'est le lien "contribute to softgoal". Ce lien représente un "goal", "task", "resource" ou "softgoal" qui peut contribuer positivement ou négativement à la réalisation du "softgoal" sans assurer cette réalisation.

On dénombre quatre types de lien "contribute to softgoal" :

Softgoal-Task link

L'exécution d'une "task" contribue positivement ou négativement au "softgoal". Ici dans l'exemple, on prend quelqu'un qui achète du matériel de bureau chez IKEA et le monte lui-même.

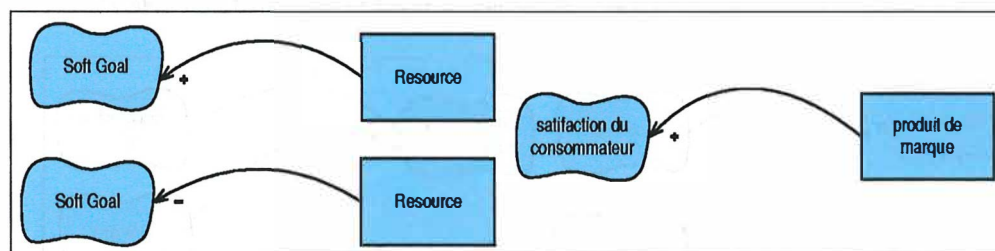
FIG. 3.19 – Représentation d'un "Softgoal-Task link" dans le modèle i*



Softgoal-Resource Link

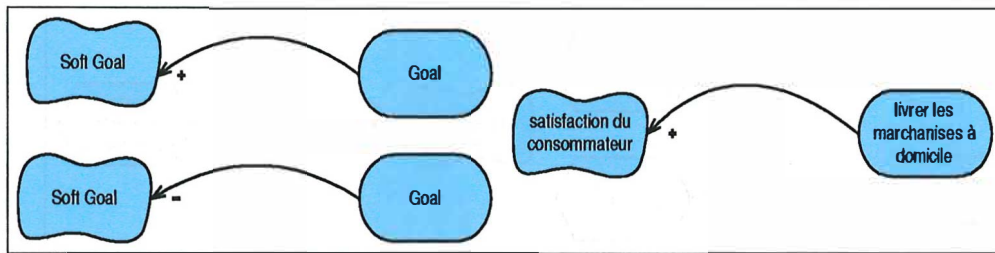
Une "resource" peut contribuer positivement ou négativement à un "softgoal". Dans l'exemple ci-dessous un produit de marque peut augmenter la satisfaction d'un consommateur.

FIG. 3.20 – Représentation d'un "Softgoal-Resource link" dans le modèle i*



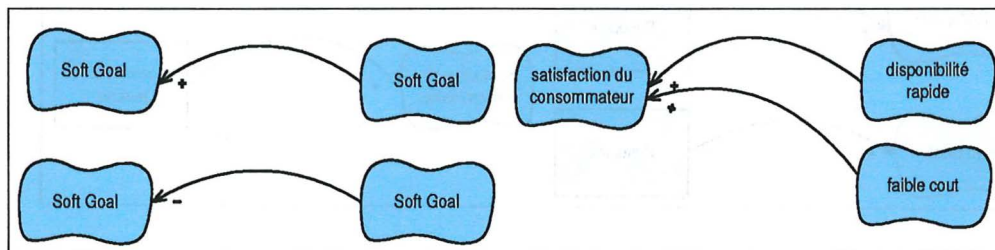
Softgoal-Goal Link

L'accomplissement d'un "goal" peut contribuer positivement ou négativement à un "softgoal". Ce lien peut être très utile dans une analyse quand il y a plusieurs alternatives. Dans l'exemple ici, un magasin peut livrer les marchandises à domicile. Cela augmente la satisfaction du consommateur.

FIG. 3.21 – Représentation d'un "Softgoal-Goal link" dans le modèle $\mathcal{S}^*$ 

Softgoal-Softgoal Link

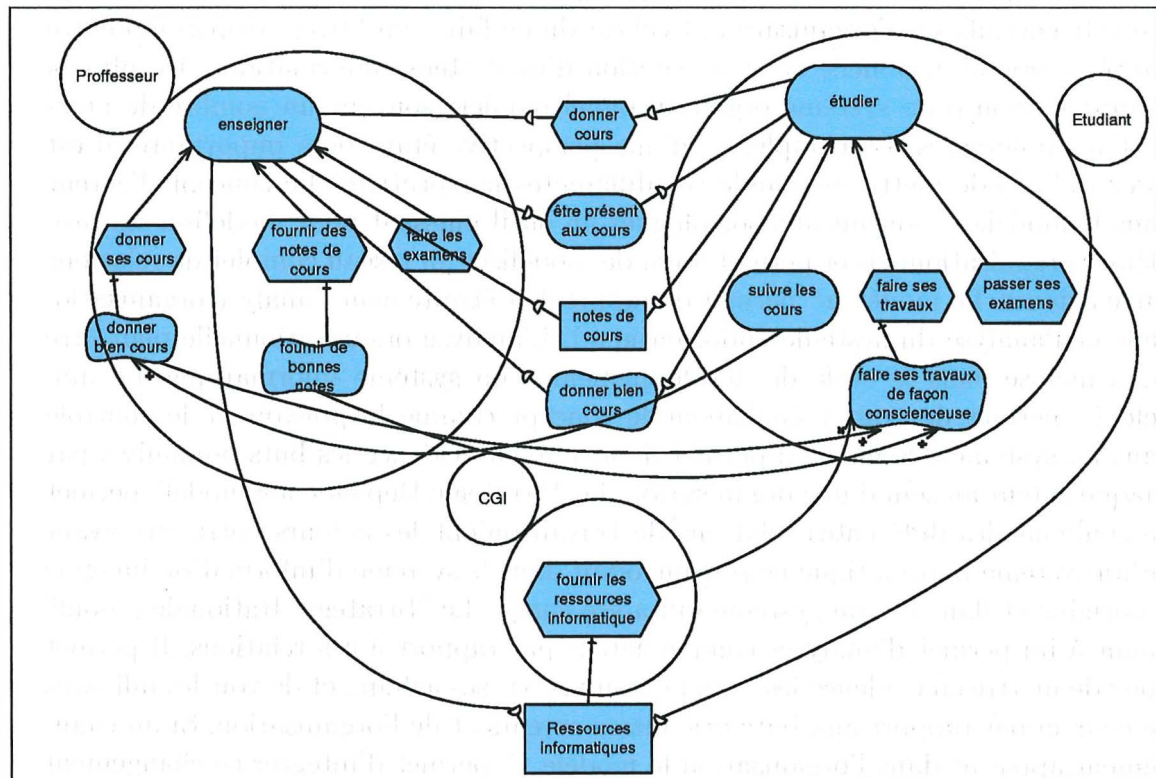
Le lien "Softgoal-Softgoal" permet le développement d'une hiérarchie de type "means-end" mais pour les "softgoals". Ce lien fournit donc une relation pour décomposer les "softgoals". Par exemple la satisfaction d'un consommateur peut être décomposée en faible coût et disponibilité rapide.

FIG. 3.22 – Représentation d'un "Softgoal-Softgoal link" dans le modèle $\mathcal{S}^*$ 

3.4.4 Exemple

Cet exemple se base de nouveau sur le système d'information de l'institut d'informatique à Namur.

FIG. 3.23 – Strategic Rationale Model



3.5 Que peut apporter le modèle i^* aux théories des organisations ?

Bien qu'un système d'information soit habituellement conçu pour accomplir certains objectifs organisationnels comme la productivité ou le service, ce système d'information une fois réalisé est souvent bien différent de celui imaginé au départ. De nombreux systèmes n'obtiennent pas les résultats attendus ou ont de sérieux impacts négatifs sur l'organisation. Ceci est dû en fait à un "trou" trop grand entre l'analyse organisationnelle et la conception d'un système informatique. De plus la compréhension d'un système organisationnel requiert souvent un nombre de perspectives différentes. La complexité d'une perspective étant déjà importante, il est assez difficile de mettre ensemble ces différentes perspectives. Le concept d'acteur dans le modèle i^* fournit aux sociologues un outil puissant pour modéliser les problèmes organisationnels et permet aussi de modéliser un réseau complet de relations entre acteurs. Le modèle i^* permet de rapprocher étroitement l'analyse organisationnelle et l'analyse du système (informatique). L'analyse organisationnelle peut être alors incluse dans le cycle de développement d'un système informatique. Le modèle i^* permet de traiter facilement le concept comme le pouvoir et le contrôle dans un système. De même il permet de mettre en évidence les buts poursuivis par chaque acteur au sein d'une organisation. Le "Strategic Dependency model" permet de souligner les différentes relations de travail qu'ont les acteurs entre eux avant qu'un système informatique ne soit introduit dans le système d'information imaginé à l'origine et dans le vrai système qui a été conçu. Le "Strategic Rationale model" quant à lui permet d'analyser chaque acteur par rapport à ces relations. Il permet donc de mettre en évidence les buts poursuivis par ses acteurs et de voir les influences de ceux-ci par rapport aux buts des autres acteurs et de l'organisation. Si un changement apparaît dans l'organisation le modèle i^* permet d'intégrer ce changement sans pour autant reconcevoir entièrement le système d'information.

Malheureusement ce modèle souffre de quelques limitations. En effet un modèle souligne nécessairement certains aspects en omettant d'autres. Tout d'abord, le modèle i^* ne fournit pas la même richesse que les méthodes classiques existantes pour décrire les organisations (textes,...) et n'a d'ailleurs pour but de les remplacer. Il permet seulement de faire le lien entre l'analyse organisationnelle et la conception du système informatique (c-à-d en gros le lien entre les sociologues et les informaticiens). Ensuite la notion d'acteur dans le modèle i^* est une notion générale. Le "Strategic Dependency model" indique seulement les relations externes entre acteurs mais n'impose aucune contrainte à ces acteurs. Enfin il ne rend compte uniquement que des dépendances entre acteurs.

Analysons maintenant plus en détails en quoi le modèle i^* permet de rapprocher les théories des organisations à l'analyse conceptuelle d'un système.

3.5.1 Les acteurs dans un organisation

La notion bien que générale d'acteur défini dans le modèle i^* peut être utilisée pour modéliser les acteurs de l'organisation. Tous les acteurs sont considérés sur un même pied d'égalité. Il n'y a pas moyen de différencier un acteur externe d'un acteur interne.

3.5.2 Les buts

Le modèle i^* permet de montrer comment les buts poursuivis par chaque acteur peuvent coexister dans l'organisation. Les concepts comme "goal", "softgoal" et "task" peuvent être employés pour modéliser ces buts. Le modèle i^* permet non seulement de montrer comment les différents acteurs réalisent les buts de mission et/ou de système de l'organisation qui sont propres à ces acteurs mais aussi comment ceux-ci s'organisent avec les buts spécifiques recherchés par chaque acteur. Le modèle n'offre cependant pas la possibilité de nuancer entre la notion de buts de mission et/ou de système recherchés par l'organisation. Il s'agit uniquement ici des buts spécifiques à chaque acteur. La notion de système de buts intégrés apparaît dans le "strategic rationale model" avec la notion de buts décomposés en sous-buts grâce aux liens "means end", "contribute to softgoal" et "task decomposition".

3.5.3 Le pouvoir

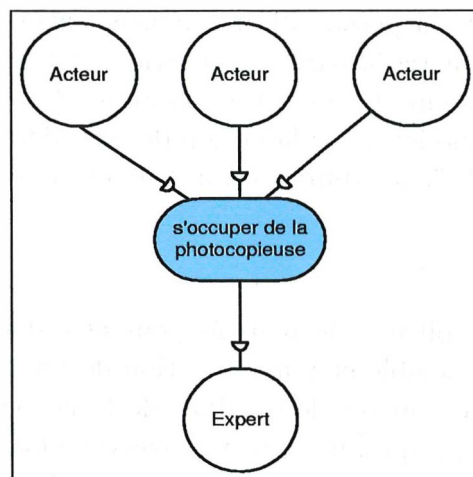
Comme on l'a vu au chapitre 2, le pouvoir peut être défini en termes de dépendances et la dépendance semble être une fonction de l'importance de ce qu'un acteur reçoit d'un autre¹. Le pouvoir dérive donc de la possession d'une ressource qu'un acteur a besoin et du fait qu'il n'y a peu ou pas de solutions alternatives pour cette ressource. La notion de dépendance entre acteurs définie dans le modèle i^* peut être utilisée pour déceler les acteurs qui ont du pouvoir. Le modèle i^* permet de mettre clairement en évidence les différentes sources de pouvoir non légitimes. En effet, que ce soit l'expertise, le contrôle de l'information ou les relations avec l'environnement, les dépendances entre acteurs pourraient permettre de mettre en évidence ces concepts et déceler ces sources de pouvoir non légitimes. Par exemple, on peut voir dans l'exemple précédent que la secrétaire académique détient une information (horaires) dont dépendent les étudiants et les professeurs et que sans cette information essentielle, ils ne peuvent accomplir leurs tâches respectives. L'expertise et le contrôle de l'information peuvent directement être perçus dans le modèle i^* . En ce qui concerne la relation avec l'environnement le modèle i^* tel qu'il est défini ne permet pas toujours de déceler directement ce problème, étant donné que i^* rend compte uniquement des relations entre les acteurs à l'intérieur d'un système. Quant à l'usage des règles comme source de pouvoir le modèle i^* ne semble pas, à première vue, très adéquat pour détecter ce problème, étant donné de nouveau la notion d'acteurs exerçant des objectifs.

¹[Pfe81]

L'expertise

L'expertise tient en la détention d'un savoir unique et crucial pour le fonctionnement de l'organisation. L'expert est le seul à pouvoir résoudre un certain nombre de problèmes fondamentaux de l'organisation qui, si ils ne sont pas résolus, constituent pour cette dernière une zone d'incertitude. Les "dependency link" permettent de mettre en évidence les acteurs qui détiennent une expertise sur une "resource". En effet un acteur ("dependee") a une expertise sur un "dependendum" si on observe un nombre assez important de "dependency link" provenant des autres acteurs ("dependers"). Ici le schéma ci-dessous illustre l'exemple du cours de sociologie dans lequel un employé est le seul à maîtriser la photocopieuse dans les moindres détails et donc tous les autres sont dépendants de lui.

FIG. 3.24 – L'expertise dans le modèle i^*



Dans l'exemple précédent on peut voir que le CGI a l'expertise des moyens informatique vu que tous les autres acteurs dépendent de lui.

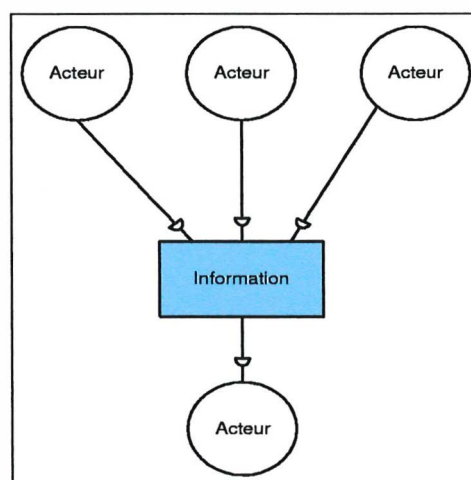
Le contrôle et l'usage de l'information

Toute organisation a comme "matière première" l'information". Il faut que celle-ci circule, qu'elle irradie l'organisation en temps et lieux voulus pour empêcher tout dysfonctionnement.

Or par la position ou la place occupée dans la structure de l'organisation, certains individus se présentent comme des maillons ou des relais importants de cette information. Ils peuvent user de cette position pour construire leur pouvoir et négocier certains avantages dans les relations qu'ils nouent avec les autres.

Dans le modèle i^* , cette source de pouvoir peut être mise en évidence comme pour l'expertise par le fait que plusieurs "dependers" dépendent d'un "dependee" pour un "dependum". Le "dependum" constituant l'information est généralement matérialisé dans le modèle i^* par une "resource".

FIG. 3.25 – Le contrôle et l'usage de l'information dans le modèle i^*



L'exemple de la secrétaire académique dans le modèle précédent illustre parfaitement ce concept.

Chapitre 4

Les outils logiciels pour le modèle i^*

Deux logiciels ont été développés pour le modèle i^* . Ils s'agit d'outils permettant de représenter graphiquement les deux modèles i^* à savoir le "strategic dependency model" et le "strategic rationale model". Ces deux outils s'appellent respectivement Redepend et OME.

4.1 Redepend

Redepend pour "Requirements Dependencies" est un outil développé par le "Center for Human Computer interaction design"¹ qui fait partie de la "City University" à Londres. Cet outil a été développé dans le cadre de deux de leurs projets incluant le modèle i^* à savoir le projet "eurocontrol", un projet à l'échelle européenne, qui a pour objectif de gérer le contrôle du trafic aérien en Europe. La partie qui intègre le modèle i^* s'appelle CORA-2 (Conflict Resolution Assistant)². L'autre projet se nomme le "SIMP project" (Systems Integration for Major Projects)³. La partie dans lequel intervient i^* s'appelle le "Pattern Language". Il s'agit en gros d'un langage prototype de modélisation de système de manoeuvres de sous-marins. Ce langage permet de créer des scénarios basés sur des analyses des exigences et de l'architecture d'un système. Il produit un nouvel environnement modélisé qui permet de mettre en évidence les dépendances entre les exigences (requirements) d'un système, son architecture et ses composants.

Redepend constitue un "template" développé en "visual basic" se basant sur le moteur du logiciel "Microsoft Visio"⁴. Cet outil intègre les deux modèles présentés dans le modèle i^* plus des outils bien spécifiques aux projets cités plus haut. Je vais

¹De plus amples informations peuvent être trouvées ici : http://www-hcid.soi.city.ac.uk/about_the_centre.htm

²De plus amples informations à propos de ce projet peuvent être trouvées ici : <http://www-hcid.soi.city.ac.uk/CORA-2.htm>

³Une explication très détaillée de ce projet peut être trouvée ici : <http://www.dcs.qmul.ac.uk/~simp/>

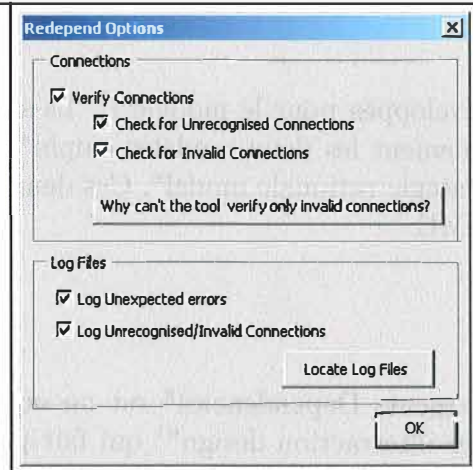
⁴<http://www.microsoft.com/office/visio/>

me limiter par la suite à expliquer uniquement les fonctions utiles pour le modèle i^* en omettant les fonctions propres aux projets expliqués ci dessus.

4.1.1 Lancement de l'outil

Pour exécuter Redepend il faut lancer Microsoft visio puis creer un nouveau document et choisir le type Redepend. Une fois ceci fait, on obtient un écran d'accueil demandant si on veut activer la gestion d'erreurs et la vérification des connexions.

FIG. 4.1 – la fenêtre d'accueil de Redepend



4.1.2 l'interface

L'interface se présente en deux volets :

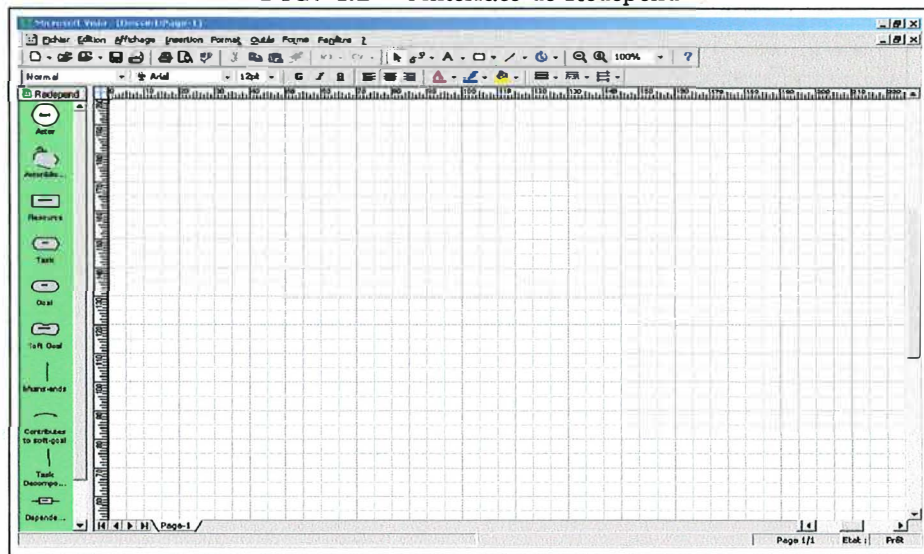
- le volet gauche contient tous les éléments utilisés dans le modèle i^*
- le volet de droite contient un panneau blanc dans lequel on peut placer ces éléments et créer les modèles.

4.1.3 Utilisation

Les éléments du modèle i^*

Pour creer un modèle il suffit de sélectionner les éléments à l'aide de la souris et de les faire glisser sur le panneau. Pour ajouter un "goal" par exemple, on selectionne un goal puis on le fait glisser sur le panneau. Une fois placé, on peut le renommer en double-cliquant dessus. On peut aussi ajuster sa taille en cliquant dessus et en selectionnant une des poignées qui constituent la figure.

FIG. 4.2 – l'interface de Redepend



Les liens

Une fois les figures placées on doit les lier ensemble à l'aide des liens mentionnés dans le chapitre 2. Pour ce faire il suffit de glisser le lien choisi sur le panneau et le lier à un point d'attache. Une fois lié à ce point l'extrémité devient rouge comme montré dans l'image ci-dessous.

FIG. 4.3 – Ajout d'un "goal"

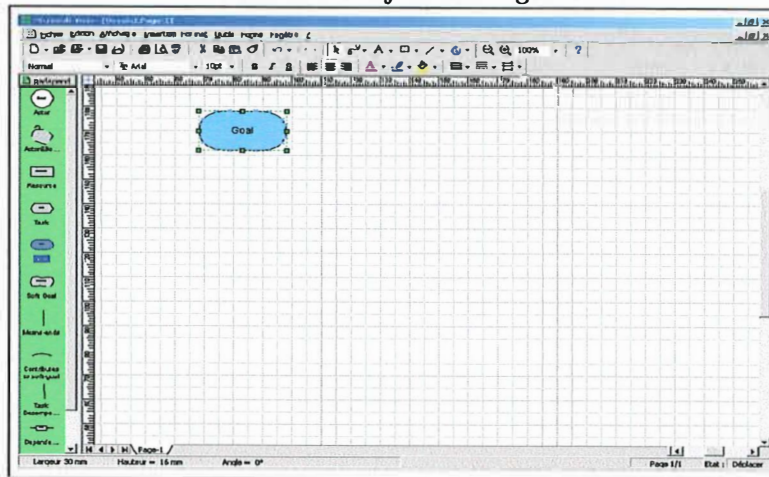
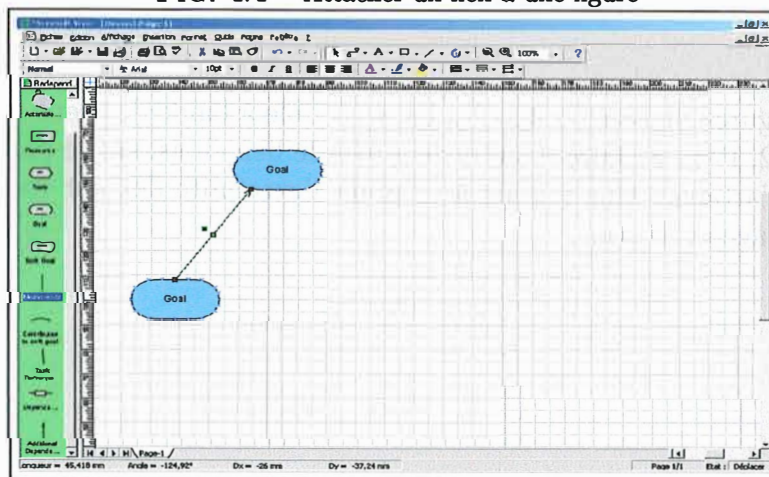


FIG. 4.4 – Attacher un lien à une figure



L'outil "redepend" permet de vérifier la connectivité des liens avec les figures, par exemple si une "task décomposition" a bien pour extrémité une "task".

FIG. 4.5 – Message d'erreur lors d'une liaison incorrecte

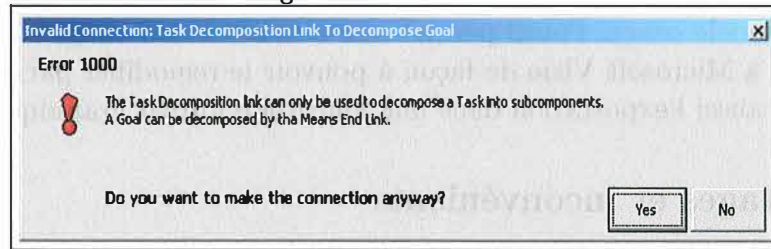
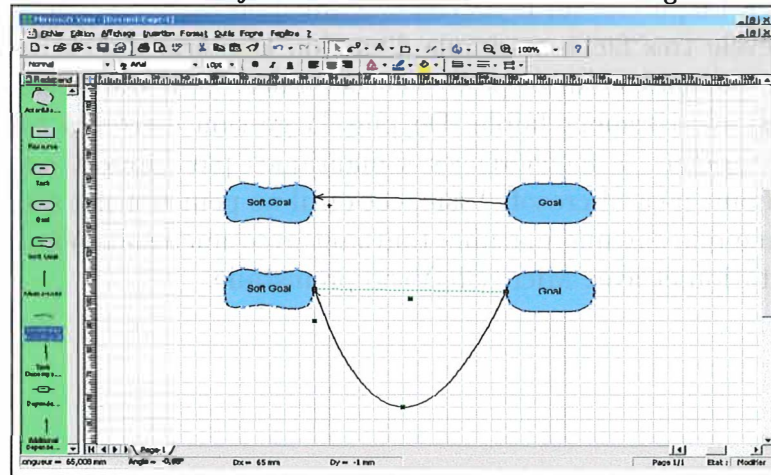


FIG. 4.6 – Ajustement du lien "contribute to softgoal"



Pour les liens "contribute to softgoal" et "dependency" une fois ces liens placés, on peut en modifier la forme en l'inclinant par exemple. Pour ce faire il suffit de sélectionner le lien en cliquant dessus puis de sélectionner une poignée et l'étirer dans le sens désiré. Pour les "dependency link", on peut aussi changer la direction du lien par un click droit sur celui-ci puis en sélectionnant "change direction" dans le menu déroulant.

4.1.4 Enregistrement

Une fois le modèle conçu, l'outil permet non seulement un enregistrement dans le format propre à Microsoft Visio de façon à pouvoir le remodifier par après, mais encore il permet aussi l'exportation dans une multitude format graphique.

4.1.5 Avantages et inconvénients

Le principal avantage de cet outil est qu'il utilise le moteur de Microsoft visio et donc peut s'intégrer parfaitement à la suite de Microsoft Office. De plus il permet de sauver les modèles créés en de nombreux formats de fichier comme par exemple le format "jpeg" pour les publier sur le web ou l'inclure dans un document. L'utilisation de cet outil se révèle très facile et simple d'emploi. L'interface y est très intuitive et il ne faut que quelques minutes pour bien la cerner. La distance sémantique et articulatoire y est relativement faible. De plus la gestion d'erreurs est incluse et donc permet de repérer les incohérences sémantiques par rapport au modèle i^* . D'un autre côté étant lié à Microsoft Visio, cet outil ne peut tourner sans celui-ci et donc ne fonctionne que sur une machine tournant avec microsoft Windows comme système d'exploitation et avec Microsoft Visio préinstallé.

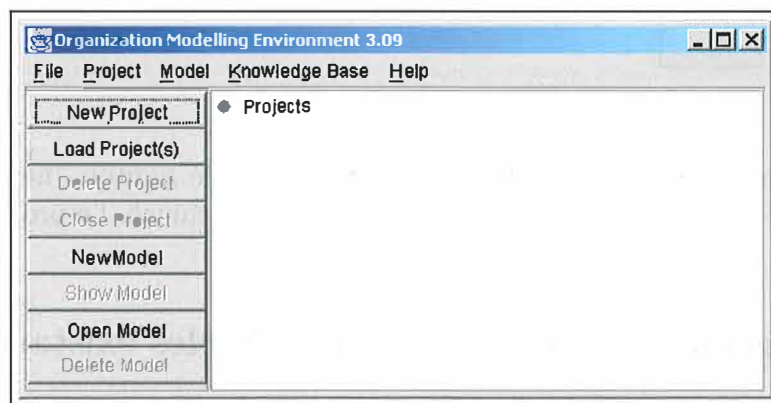
4.2 OME 3

OME pour Organization Modelling Environment est un outil d'analyse et de modélisation orienté agent/but. Il supporte à la fois le modèle i^* et *NFR*. Ici je vais juste m'étendre sur la fonction qu'offre ce logiciel pour le modèle i^* .

4.2.1 Lancement de l'outil

Pour lancer l'outil vous devez disposer d'une machine virtuelle Java⁵ sur votre ordinateur. C'est la seule contrainte. Une fois lancé, on obtient la fenêtre principale

FIG. 4.7 – Fenêtre principale d'OME3



d'OME. Dans cet outil les modèles i^* sont regroupés en projet. A partir de cette fenêtre vous pouvez soit créer un projet, soit en charger un existant.

⁵<http://java.sun.com/>

4.2.2 Création d'un projet

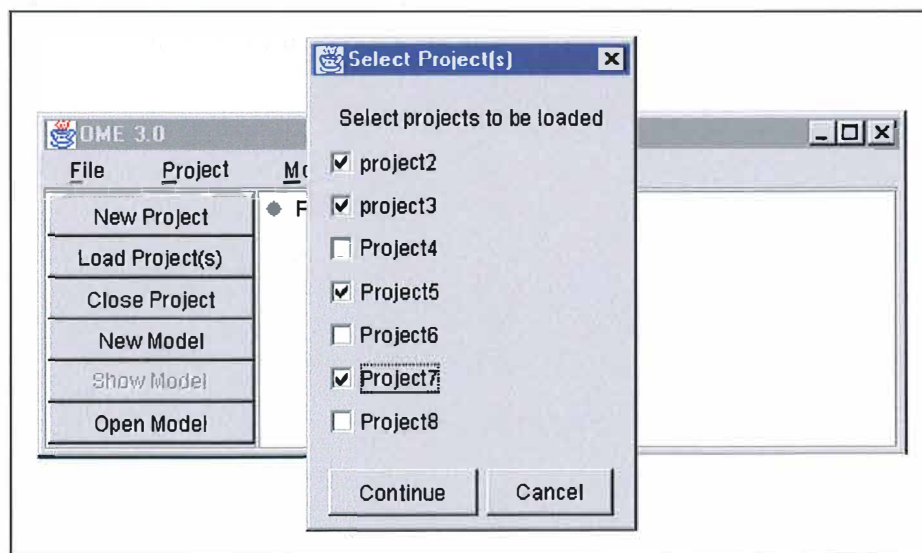
FIG. 4.8 – Création d'un projet



Pour créer un nouveau projet il suffit de cliquer sur le bouton "new project". Après avoir entré un nom de projet il faut cliquer sur "continue". Le projet est ainsi créé.

4.2.3 Chargement d'un ou plusieurs modèles existants

FIG. 4.9 – Chargement d'un projet



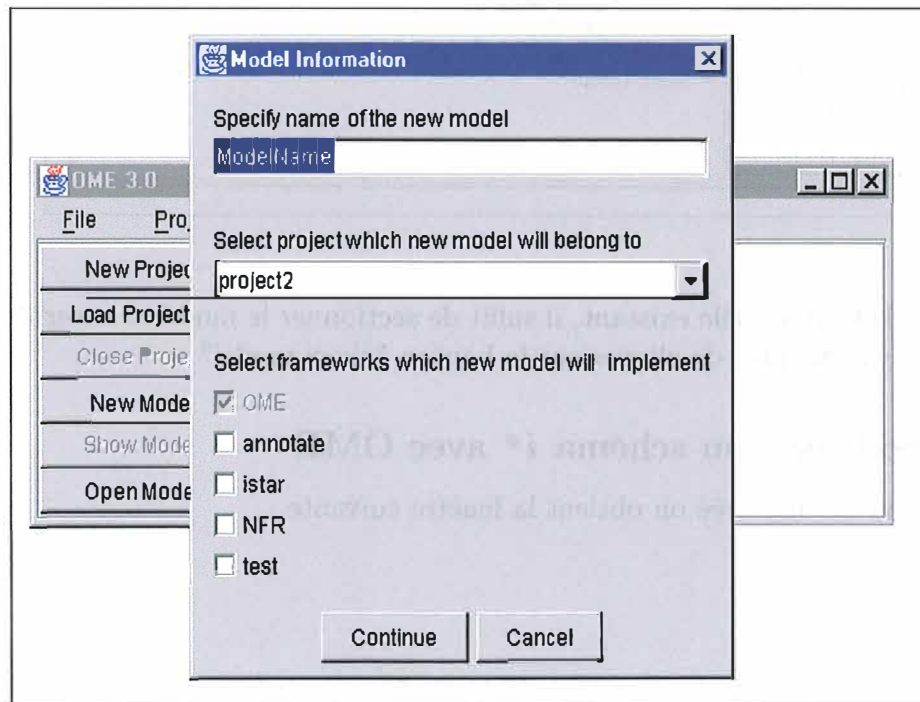
Pour charger les projets existants il suffit de cliquer sur le bouton "load project(s)" puis sélectionner le ou les projets à charger.

4.2.4 Fermeture d'un projet ouvert

Pour fermer un projet ouvert il suffit de sélectionner le projet dans la fenêtre de gauche puis cliquer sur le bouton "close project".

4.2.5 Création d'un nouveau modèle

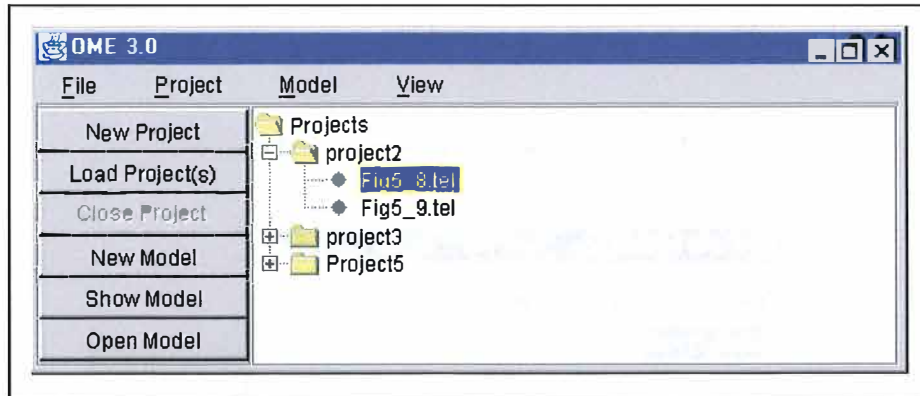
FIG. 4.10 – Création d'un nouveau modèle



Une fois un projet chargé, on a la possibilité de créer un ou plusieurs modèles à l'intérieur de celui-ci. Pour ce faire il suffit de cliquer sur le bouton "new model". Une fois ceci fait on obtient la fenêtre ci-dessus. Il faut alors entrer le nom du modèle, sélectionner le projet dans lequel on veut l'inclure et enfin sélectionner le type de modèle. Dans notre cas, il faut sélectionner "istar". Quand tout ceci est fait on clique sur "continue" et on obtient l'outil de création de modèle.

4.2.6 Afficher un modèle existant

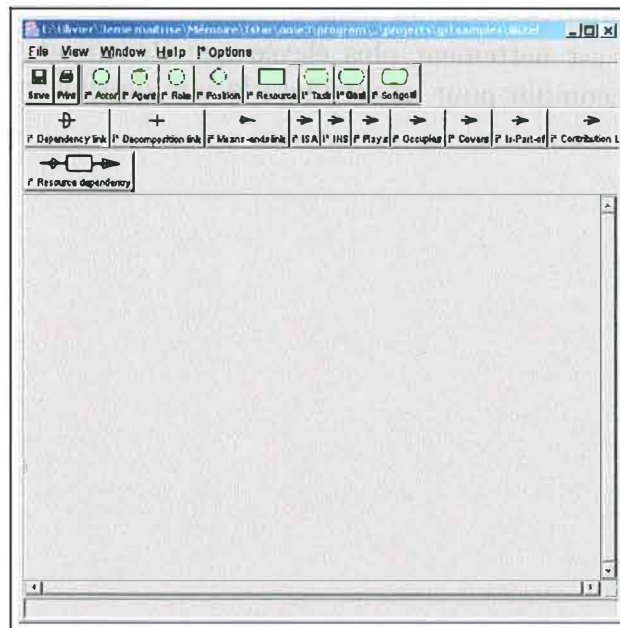
FIG. 4.11 – Afficher un modèle existant



Pour afficher un modèle existant, il suffit de sectionner le modèle concerné dans la fenêtre de gauche puis de cliquer sur le bouton "show model".

4.2.7 Modéliser un schéma i^* avec OME

Une fois un modèle créé on obtient la fenêtre suivante :

FIG. 4.12 – Modéliser un schéma i^* 

Pour ajouter un élément i^* dans le panneau principal il suffit de selectionner un des éléments en cliquant dessus dans la barre des outils situés en dessous de la barre des menus. Pour placer cet objet dans le panneau central il suffit de cliquer à l'endroit voulu.

Pour placer un lien i^* entre deux éléments du modèle il faut sélectionner le lien dans la barre des outils puis cliquer sur les deux éléments à relier qui sont déjà placés sur le panneau central.

Une fois les éléments placés on peut bien sûr les redimensionner, les renommer, les effacer, ou encore changer la direction s'il s'agit de liens. Pour ce faire il faut selectionner l'élément puis activer avec le bouton droit de la souris le menu déroulant. On y trouve les options que j'ai cités plus des options spécifiques à chaque élément.

Une fois le modèle terminé, on peut l'enregistrer dans le format spécifique à OME pour une modification ultérieure. Une autre option intéressante est d'exporter ce schéma dans le format graphique png pour l'inclure dans d'autres documents. Cette fonction se trouve dans le menu file puis export.

4.2.8 Avantages et inconvénients

Un des principaux avantages de OME c'est qu'il n'est pas dépendant d'un logiciel tiers par rapport à Redepend. De plus il a été développé en java ce qui permet une

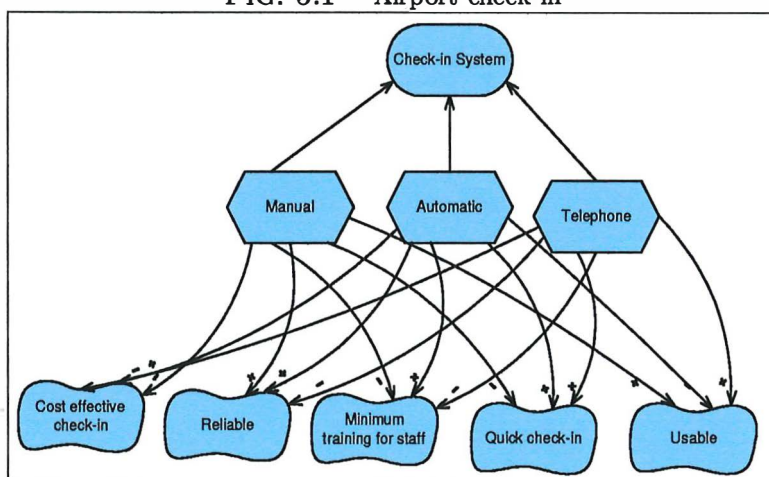
portabilité aisée sur n'importe quelle plateforme. Il offre aussi la possibilité de créer d'autres modèles de la même façon notamment des modèles *NFR*. Cependant il exige d'importantes ressources systèmes pour fonctionner. La distance sémantique et surtout articulatoire est nettement plus élevée que Redepend étant donné les nombreuses actions à accomplir pour lancer l'interface et dessiner un modèle.

Chapitre 5

Possibles améliorations du modèle i^*

Comme nous l'avons vu le modèle i^* est un outil puissant de modélisation qui permet de rapprocher l'analyse organisationnelle de la conception de système d'information. Malgré tout ce modèle souffre de quelques faiblesses. En effet regardons de plus près l'exemple ci dessous :

FIG. 5.1 – Airport check-in



Ce schéma est la traduction d'un système de verification de billets dans un aéroport. Il y a trois systèmes différents à savoir un système manuel (manual), automatique (automatic) et par téléphone (telephone). Le premier système est un système classique dans lequel on fait la file devant un guichet, le deuxième consiste quant à lui de remplacer le guichet par un ordinateur. Le troisième système est assez original par le fait qu'on téléphone à la compagnie aérienne. Un motocycliste se rend alors directement chez vous pour la validation des billets. On aimerait bien savoir laquelle des trois solutions est la meilleure. Le modèle i^* tel qu'il l'est défini ne peut répondre à cette question. C'est pourquoi j'ai introduit une méthode d'analyse

multicritère dans ce modèle. Elle s'appelle Analytic Hierarchy Process(AHP)¹.

Un autre point dont le modèle *i** fait défaut sont les attributs et les conditions. En effet des modélisations comme UML en sont pourvues mais ici ces concepts sont absents. pourtant ils pourraient être utiles pour caractériser et nuancer certains aspects.

¹[Saa80]

5.1 The Analytic Hierarchy Process(AHP)

"It is better to be approximately right than precisely wrong", Thomas L. Saaty

Les managers font face chaque jour à des décisions complexes qui sont interconnectées et dependent l'une de l'autre. Il est pour la plupart impossible de faire de tel décisions en se basant uniquement sur l'intuition et la logique. La prise de décision a pour objectif de trouver la " meilleure " alternative ou un ensemble d'alternatives en considérant un certain nombre d'objectifs, de critères et d'autres facteurs importants. La méthode AHP est utilisée pour aider un décideur à extraire la complexité d'un problème de décision et à se servir de ses jugements. Une décision complexe basée sur un certain nombre de critères exige des compromis entre les différents critères afin de pouvoir tirer la meilleure alternative de choix. Même un problème simple peut ne pas aboutir à une solution rapide. Parfois, les décisions sont prises en satisfaisant, parmi la liste des critères considérés, un critère à chaque fois. Il est mis au point des alternatives permettant de satisfaire le critère le plus important et on élimine les alternatives qui restent. Il est considéré un deuxième critère moins important que le premier et plus important que les autres, et il est sélectionné, parmi les alternatives déjà prises au premier tour, celles qui satisfont ce second critère, et ainsi de suite. Cette démarche peut ne pas aboutir à la meilleure alternative. Ce dont on a besoin c'est une méthode de décision pouvant utiliser des structures qui, d'une part, représentent un problème de décision et, d'autre part, développent des priorités pour les alternatives; ces priorités doivent être basées sur les jugements individuels apportés par les opérateurs intervenant au niveau de la prise de décision, et ceci tout au long du processus de décision.

AHP a été appliquée pour la première fois par Thomas L. Saaty en 1971 pendant qu'il travaillait sur des problèmes de planification d'urgence pour le Ministère de la Défense aux Etats-Unis. La maturité de l'application de cette théorie a été confirmée sur plusieurs études (transport au Soudan en 1973, Estimation des distances, Psychothérapie, Allocation d'énergie, Sélection d'entrepreneurs, ...etc.). Je vais présenter dans ce qui suit le principe de la méthode AHP puisque c'est sur cet algorithme que se base l'amélioration du modèle i^* que je propose.

AHP décompose un problème complexe de décision en un ou plusieurs niveaux de détails où l'évaluation des valeurs est fournie par des comparaisons par paires. Dans le but de fournir une approche structurante, des vecteurs de priorités sont établis en utilisant des mesures de cohérence. La méthode AHP est donc un processus de mesure à l'intérieur de structures ou de réseaux hiérarchiques, permettant à un individu d'exprimer ses préférences parmi plusieurs critères et alternatives en ce qui concerne chaque critère et ensuite, d'obtenir un classement d'ensemble des alternatives utilisant les poids des critères. Pour cela, des comparaisons relatives, des structures de normalisation ainsi que des fonctions permettant de mesurer non

seulement les priorités entre les critères mais aussi d'étudier l'influence mutuelle des différents niveaux hiérarchiques, sont clairement définies. AHP est une méthode relativement simple à appliquer et à comprendre. Elle permet de :

- prendre en compte les critères qualitatifs et les critères quantitatifs,
- décomposer un problème complexe et l'examiner en multiples niveaux,
- mesurer la cohérence des comparaisons effectuées par le décideur.

5.1.1 Présentation intuitive de la méthode AHP

AHP est un outil puissant et flexible de prise de décision pour des problèmes multicritère complexes pour lesquels des aspects *qualitatifs* et *quantitatifs* doivent être incorporés. Elle a été bien reconnue dans le passé comme étant un outil puissant dans la prise de décision et pour la planification. La méthode AHP aide les décideurs à structurer les composantes importantes d'un problème en une structure hiérarchique semblable à une arborescence. Puis, les résultats sont synthétisés en ramenant des décisions complexes à une série de simples comparaisons et de rangements. La méthode AHP permet d'aboutir à la meilleure décision en fournissant un raisonnement clair pour le choix effectué. Ayant été conçu pour refléter la voie de penser des personnes, la méthode AHP a été développée il y a plus de 20 ans et continue à être la théorie de prise de décision la plus largement répandue et utilisée.

AHP est donc un processus de décision qui interprète directement les données et les informations en formant des jugements sur lesquels on exécute une échelle de mesure à l'intérieur d'une structure hiérarchique.

5.1.2 Démarche à suivre

La démarche s'effectue en quatre phases :

Construction d'une structure hiérarchique pour le problème de décision

Une hiérarchie est un type particulier de systèmes basé sur les hypothèses suivantes :

- Les entités identifiées peuvent être groupées dans des ensembles disjoints, appelés groupes ou encore niveaux.
- Les entités d'un groupe B, influençant les entités d'un seul autre groupe C, sont elles-mêmes influencées par les entités d'un seul autre groupe A.
- Les éléments de chaque groupe(appelé également niveau) sont supposés être indépendants.

Les avantages d'une telle hiérarchie se résument à :

- un pouvoir *descriptif* qui montre bien comment un changement d'une priorité à un niveau supérieur peut affecter la priorité à un niveau inférieur.
- un niveau de *détails* très appréciable et un aperçu des acteurs et de leurs objectifs dans les niveaux supérieurs.
- des niveaux *flexibles* car l'addition d'indicateurs ou de critères ou même des machines à une structure bien hiérarchisée ne perturbe pas sa performance globale.

Pour structurer le problème, on commence par considérer tous les éléments du problème, des plus généraux aux plus spécifiques. Il s'agit ensuite de les ranger en niveaux tout en suivant les directives suivantes :

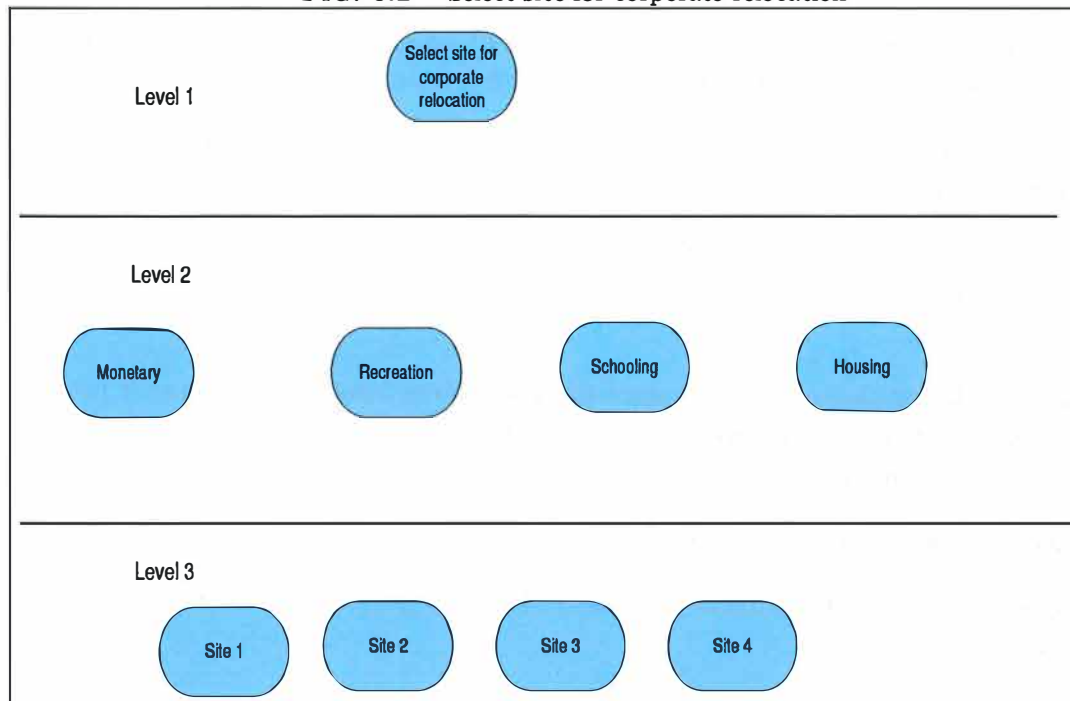
- niveau 1 : identifier les objectifs généraux.
- niveau 2 : identifier les critères de décision (sous-objectifs) qui satisfont chaque objectif.
- niveau 3 : identifier les alternative.

Exemple : choisir un site pour placer une industrie

Une entreprise souhaite choisir un site approprié pour ses activités. Elle a le choix parmi quatre sites. Les critères de décision sont en anglais Monetary, Recreation, Schooling et Housing. Dans cet exemple on va essayer de voir quel est le meilleur choix.

La première étape, comme on l'a vu plus haut, consiste à développer une hiérarchie de trois niveaux.

FIG. 5.2 – Select site for corporate relocation



Comparaison par paire des éléments de chaque niveau de la hiérarchie construite

Une comparaison par paire, appelée "pair-wise comparaison" par les anglophones, évalue l'importance relative de deux éléments vis-à-vis de l'opérateur (ou décideur) contribuant à la réalisation de l'objectif du niveau supérieur adjacent. Un décideur compare donc des éléments d'un même niveau (par exemple, les critères), par paire, ce qui lui permet d'établir ses préférences. Pour cela, il faut choisir une échelle de valeurs pour spécifier le degré d'importance d'un élément par rapport à un autre. Nous adoptons l'échelle de valeur (1-9) (également appelée échelle de Saaty) utilisée dans la méthode AHP. Dans le but de mieux comprendre les comparaisons exploitées dans l'algorithme, présentons l'échelle (1-9) exploitée :

Echelle de Saaty de l'importance relative		
Degré d'importance relative	Definition	Explication
1	Importance égale	2 activités contribuent de manière égale à un objectif. (aucune préférence)
3	Importance modérée	Le jugement accorde une importance légèrement modérée à un critère par rapport à un autre.
5	Fortement plus important	Le jugement suggère une forte préférence à un critère plutôt qu'à un autre.
7	Très fortement plus important	Une activité est très fortement préférée et cette préférence est démontrée dans la pratique.
9	Absolument plus important	Une préférence absolue pour un critère.
2,4,6,8	Valeurs intermédiaires	valeurs intermédiaires pour mettre en évidence des compromis.
Reciproques		Valeurs Inverses utilisées pour montrer la dominance du second élément par rapport au premier.

TAB. 5.1 – Echelle de Saaty

	Monetary	Recreation	Schooling	Housing
Monetary	1	0,33	0,11	0,11
Recreation	3	1	0,2	0,2
Schooling	9	5	1	0,5
Housing	9	5	2	1
Weight	0,04	0,1	0,36	0,5

TAB. 5.2 – The criteria Matrix

Voici un exemple pour le critère "Schooling" par rapport aux quatre sites :

	Site 1	Site 2	Site 3	Site 4
Site 1	1	0,2	0,11	0,2
Site 2	5	1	0,2	0,33
Site 3	9	5	1	8
Site 4	5	3	0,13	1
Weight	0,05	0,14	0,53	0,28

TAB. 5.3 – The alternative Matrix for schooling

La comparaison par paire étant effectuée, il faut rechercher un vecteur de priorité qui classe les alternatives par ordre croissant ou décroissant. Le classement par priorité des éléments d'un niveau de la hiérarchie contribuant à l'accomplissement d'un objectif du niveau supérieur adjacent est appelé *poids relatif* ou *ordre de priorité*.

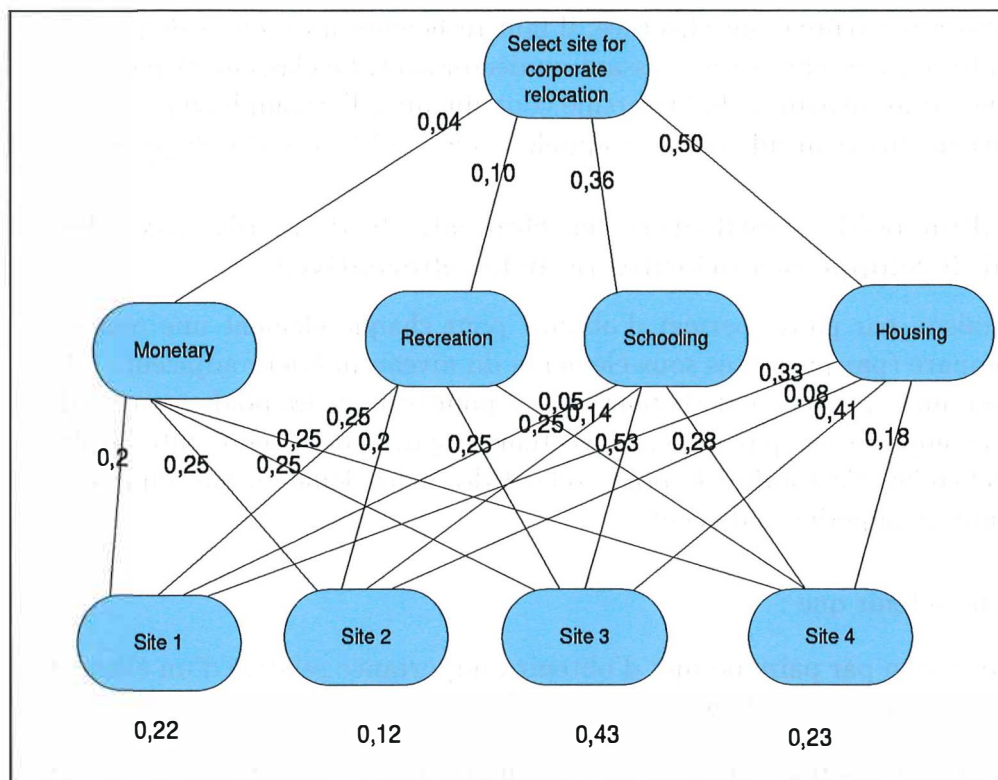
Estimation d'un poids relatif entre les éléments de deux niveaux adjacents, ce qui développe des priorités pour les alternatives

La comparaison par paire, permet d'obtenir pour chaque élément une matrice carrée, qui compare (par paire) ses sous-éléments du niveau inférieur adjacent. T.L. Saaty a proposé une approche par vecteur propre pour estimer les poids à partir de la matrice des comparaisons par paire. Le vecteur propre obtenu représente l'ordre de priorité recherché, c'est-à-dire le poids relatif des sous-éléments par rapport à l'élément du niveau supérieur adjacent.

On peut donc retenir que :

- La comparaison par paire permet d'obtenir l'importance relative d'un élément par rapport à un autre (d'un même niveau).
- Le vecteur de priorité ou poids (vecteur d'importance relative) exprime quant à lui le rangement relatif des éléments d'un niveau par rapport à chacun des éléments du niveau adjacent supérieur.

Voici donc la solution pour l'exemple ci-dessus :

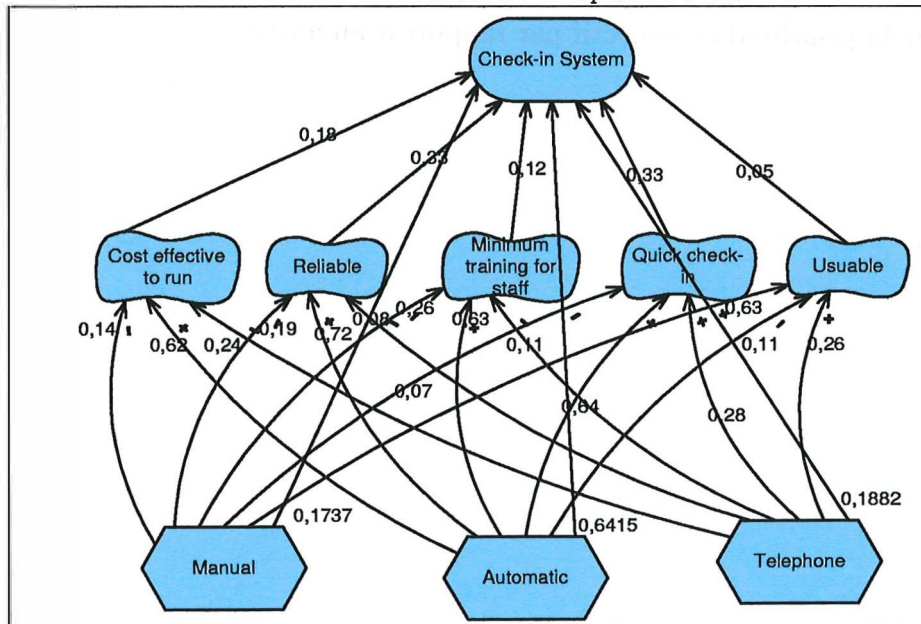


On peut donc voir que c'est le site 3 qui correspond le mieux aux attentes de l'entreprise.

5.1.3 Que peut apporter AHP au modèle i^* ?

Bien qu'utile pour modéliser des systèmes, la nature informelle du modèle i^* peut parfois le rendre difficile à utiliser pour une prise de décision plus rigoureuse et plus quantitative qui peut être essentielle pour la modélisation de système concurrent. En effet, l'utilisation du modèle i^* a parfois été critiquée pour le manque de nombres indiquant le degré d'importance d'importance de certains buts. En fait un des problèmes dans le modèle i^* est que lorsque nous avons une décomposition de but² en sous-but, on ne sait pas définir l'importance relative de ces sous-buts par rapport au but principal. Il est de même pour les tâches. Pour avoir une valeur absolue objective on peut utiliser AHP, et particulièrement la "comparaison par paire" expliquée plus haut. On peut donc considérer comme niveau 1 le but (ou la tâche) principal et comme niveau 2 les sous-buts (ou sous-tâches). On peut ensuite construire la matrice. Voici par exemple la solution du problème du check-in à l'aéroport.

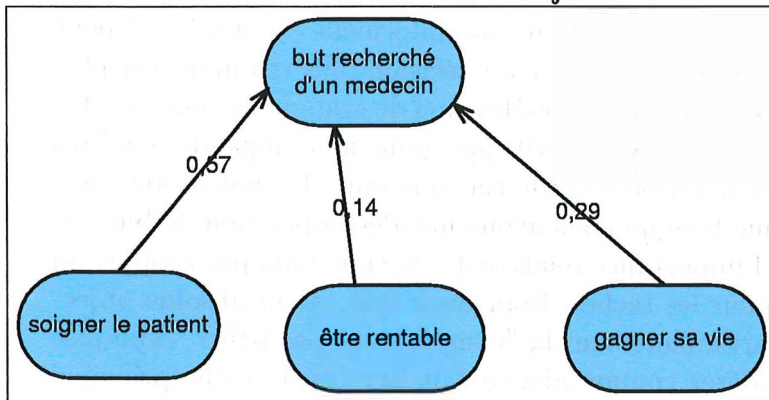
FIG. 5.3 – Airport check-in



Une autre application de AHP est de souligner le degré de contribution de chaque sous-but à un objectif et ainsi de voir quel est le sous-but le plus important. Dans l'exemple qui suit on peut voir les objectifs d'un medecin et voir quels sont ses objectifs prioritaires.

²"goal" et "softgoal"

FIG. 5.4 – Objectif d'un médecin



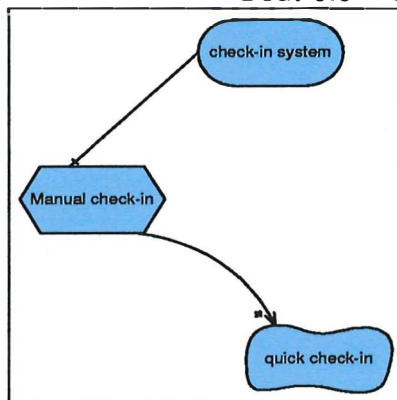
Avec cet exemple simple on peut voir directement que soigner un patient est l'objectif prioritaire de ce médecin. AHP apporte au modèle i^* la possibilité d'exprimer la contribution de chaque sous-but à un but principal et donc de montrer l'objectif prioritaire et de voir la priorité d'un objectif par rapport à un autre.

5.2 Les attributs et conditions dans le modèle i^*

5.2.1 Introduction

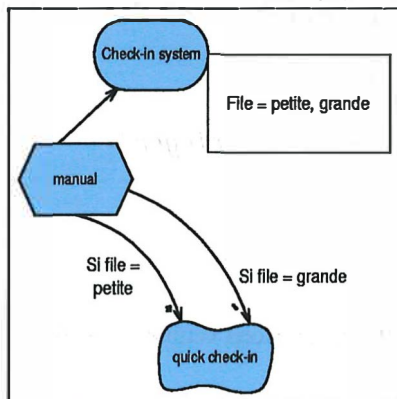
Un autre aspect faisant défaut dans le modèle i^* est celui des attributs et conditions. En effet les relations entre certains acteurs peuvent être très différentes pour un ou plusieurs attributs et/ou dans certaines conditions. Dans l'exemple qui suit nous voyons de nouveau le système de check-in. On peut voir que le système manuel nuit à la rapidité. Mais qu'en est-il si par exemple il y a seulement une petite file devant le guichet ?

FIG. 5.5 – Check-in system sans attributs et conditions



Voyons maintenant ce que cela donne si on rajoute des attributs et conditions. On peut donc rajouter un attribut file qui peut prendre deux valeurs : grande ou petite. On rajoute aussi une condition sur le lien "contribute to soft-goal". On peut donc dédoubler ce lien car il peut contribuer positivement ou négativement selon que la file est grande ou petite.

FIG. 5.6 – Check-in system avec attributs et conditions



Nous pouvons donc voir par cet exemple très simple l'utilité des conditions dans le modèle i^* .

5.2.2 Syntaxe et sémantique

Afin de valider cette amélioration j'ai développé une syntaxe et sémantique pour les attributs et les conditions.

Attributs

Pour chaque élément du modèle i^* (goal, softgoal, resource, task and actor) on peut associer différents attributs. La syntaxe que j'ai définie pour ceci est la suivante :³ :

- $\langle \text{syntaxattribute} \rangle ::= \langle \text{attributes} \rangle$

avec:

- $\langle \text{ent} \rangle ::= \text{enter}$
- $\langle \text{space} \rangle ::= \text{espace}$
- $\langle \text{number} \rangle ::= 1|2|3|4|5|6|7|8|9$
- $\langle \text{sign} \rangle ::= +|-$
- $\langle \text{zero} \rangle ::= 0$
- $\langle \text{numbers} \rangle ::= \langle \text{zero} \rangle | \langle \text{number} \rangle$
- $\langle \text{integer} \rangle ::= \langle \text{zero} \rangle | [\langle \text{sign} \rangle] \langle \text{number} \rangle \{ \langle \text{numbers} \rangle \}^*{}^4$
- $\langle \text{char} \rangle ::= a|b| \dots |z|A|B| \dots |Z| \langle \text{space} \rangle$
- $\langle \text{boolean} \rangle ::= \text{true}|\text{false}$
- $\langle \text{rightnumber} \rangle ::= \{ \langle \text{number} \rangle \}^+ | \{ \langle \text{zero} \rangle \}^* \{ \langle \text{numbers} \rangle \}^+{}^5$
- $\langle \text{real} \rangle ::= \langle \text{integer} \rangle | \langle \text{integer} \rangle, \langle \text{rightnumber} \rangle$
- $\langle \text{string} \rangle ::= \langle \text{char} \rangle | \langle \text{string} \rangle \langle \text{char} \rangle | \langle \text{string} \rangle \langle \text{integer} \rangle$
- $\langle \text{value} \rangle ::= \langle \text{string} \rangle | \langle \text{real} \rangle | \langle \text{integer} \rangle | \langle \text{boolean} \rangle$
- $\langle \text{values} \rangle ::= \langle \text{value} \rangle | \langle \text{value} \rangle, \langle \text{values} \rangle$

³ J'utilise la grammaire ABNF pour définir la syntaxe, pour plus d'informations veuillez consulter ceci : <ftp://ftp.belnet.be/packages/rfc/rfc2234.txt>

⁴ 0..N elements

⁵ 1..N elements

- $\langle attribute \rangle ::= \langle char \rangle \mid \langle attribute \rangle \langle char \rangle \mid \langle attribute \rangle \langle number \rangle$
- $\langle attributes \rangle ::= \langle attribute \rangle = \langle values \rangle \mid \langle attribute \rangle = \langle values \rangle \langle ent \rangle \langle attributes \rangle$

La sémantique est très simple à comprendre: $attribute_i = value_1, \dots, value_m$ où

- $0 < i < n$
- $value_i \neq value_{i+1}$
- le type de $value_i$ est le même que $value_{i+1}$

and où le type de $value_i$ peut être un

- string
- integer
- real
- boolean

conditions

Pour chaque lien entre deux éléments du modèle i^* on peut mettre une condition dessus: "si *condition* alors *action*". La syntaxe des conditions est la suivante:

- $\langle syntaxcondition \rangle ::= \langle conditions \rangle$

avec

- $operator ::= \mid \neq \mid < \mid > \mid \geq \mid \leq$
- $component ::= \langle goal \rangle \mid \langle softgoal \rangle \mid \langle task \rangle \mid \langle resource \rangle \mid \langle actor \rangle$ ⁶
- $\langle leftcondition \rangle ::= \langle component \rangle :: \langle attribute \rangle$
- $\langle rightcondition \rangle ::= \langle value \rangle$
- $\langle condition \rangle ::= \langle leftcondition \rangle \langle operator \rangle \langle rightcondition \rangle$
- $\langle conditions \rangle ::= \langle condition \rangle \mid \langle conditions \rangle AND \langle conditions \rangle \mid \langle conditions \rangle OR \langle conditions \rangle$

La sémantique est alors très facile à comprendre : si l'évaluation de la condition est vérifiée alors le lien peut être accompli.

⁶Pour la syntaxe cfr i^*

Cette syntaxe assez formelle permet de ne pas être ambigu quant à la compréhension des conditions exprimées.

5.2.3 Représentation graphique

Réprésenter graphiquement les attributs et conditions sans surcharger les graphes et par de là diminuer la lisibilité n'est pas une chose aisée. En effet la représentation ci-dessus a comme désavantage pour des graphes complexes de les rendre assez illisibles. Une solution assez simple à mettre en oeuvre pour les attributs serait d'implémenter dans les logiciels un menu qui permettrait d'afficher les attributs (c'est d'ailleurs déjà le cas pour "Redepend", la notion d'attribut est insérée dans le cadre du projet eurocontrol). Cela poserait problème lors de l'impression des graphes. Une autre solution serait de les mettre sur une feuille à part. Cela augmenterait la distance articuloire par le fait qu'il faudrait jongler avec d'une part le graphe d'autre part cette feuille.

En ce qui concerne les conditions le problème est le même. La solution proposée est la même que pour les attributs, à savoir l'emploi d'une seconde feuille. Pour indiquer qu'un lien est pourvu d'une condition il suffit de lui associer un indice correspondant à la condition sur l'autre feuille.

5.2.4 Conclusion

Grâce à cette notion d'attributs (identique à UML) et conditions, le modèle i^* offre encore plus de possibilités et une meilleure compréhension. En effet, les conditions permettent de nuancer les dépendances entre les acteurs ou les différents liens entre les objectifs. Les attributs permettent de caractériser davantage chaque élément du modèle. Cependant cet ajout s'éloigne du modèle initial et de sa notion de généralité et de subjectivité.

Chapitre 6

Comment enrichir le modèle i^* grâce aux théories des organisations ?

Comme nous l'avons vu au chapitre 3 le modèle i^* est un modèle puissant qui permet de faire étroitement le lien entre l'analyse organisationnelle et l'analyse de l'existant du point de vue des relations entre acteurs et leurs buts qu'ils accomplissent. Cependant ce modèle est assez limité car il permet uniquement de représenter des concepts de base sans entrer dans les détails. Le but de ce chapitre est d'essayer de montrer de possibles extensions au modèle i^* afin qu'il permette par la suite d'encore mieux intégrer les théories des organisations dans la conception de systèmes d'information.

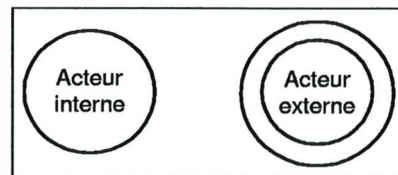
6.1 Les acteurs

Dans le modèle i^* la notion d'acteur est une notion générale. Il n'y a pas d'attributs qui permettraient de caractériser un acteur notamment de voir si c'est un acteur interne ou externe ou bien de voir sa position dans une hiérarchie.

6.1.1 Acteur interne et acteur externe

Une façon simple de différencier un acteur interne d'un acteur externe dans le modèle i^* sans l'encombrer davantage serait de modifier la forme graphique représentant l'acteur. La forme pour un acteur interne reste la même tandis que pour un acteur externe elle devient différente.

FIG. 6.1 – Acteur interne et acteur externe



Cette simple modification ne complique pas exagérément le modèle initial et ne le rend pas plus difficile à lire. Par cette modification élémentaire le modèle i^* peut maintenant différencier aisément un acteur interne d'un acteur externe. Grâce à la notion d'attributs introduits précédemment, on pourrait encore les détailler davantage, notamment en indiquant leur position dans la hiérarchie par exemple.

6.2 Les buts

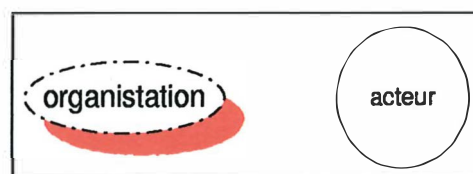
Le modèle i^* permet seulement de mettre en évidence les buts individuels poursuivis par chaque acteur. Il ne montre pas les buts poursuivis par l'organisation. Dès lors il est impossible d'exprimer avec le modèle i^* le concept de but de mission et de but de système. De même il ne permet pas de faire la différence entre les buts de système et les buts de mission.

6.2.1 Buts spécifiques et but de l'organisation

Le modèle i^* est un modèle qui permet de montrer les relations entre des acteurs. Il permet donc d'exprimer les buts individuels spécifiques recherchés par chaque acteur. Cependant il ne permet pas par contre de modéliser les buts (de mission ou de système) recherchés par l'organisation.

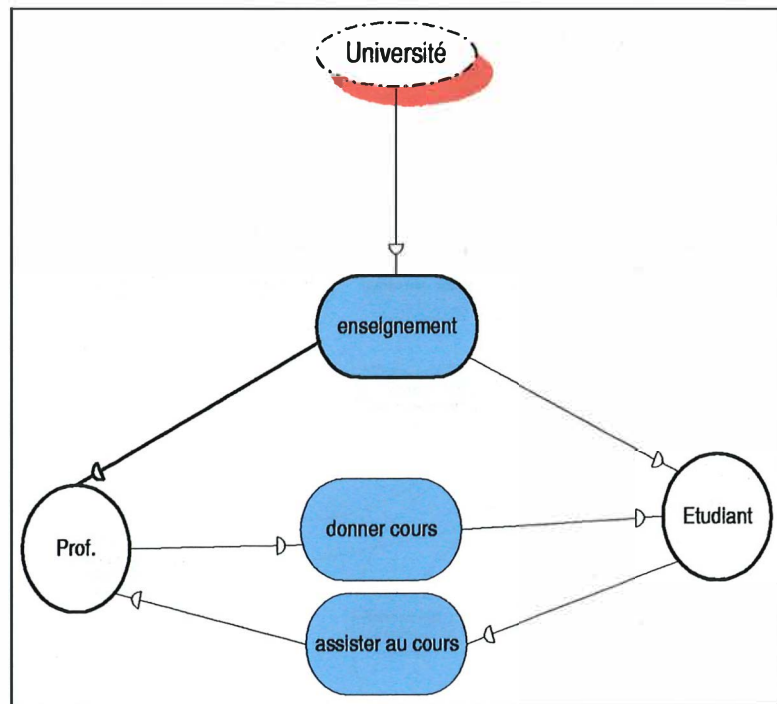
Pour résoudre ce problème on pourrait imaginer l'introduction d'un nouvel élément dans le modèle qui concrétiserait l'organisation et qui aurait les mêmes propriétés que les acteurs et qui exercerait des relations avec ceux-ci.

FIG. 6.2 – Acteur et organisation



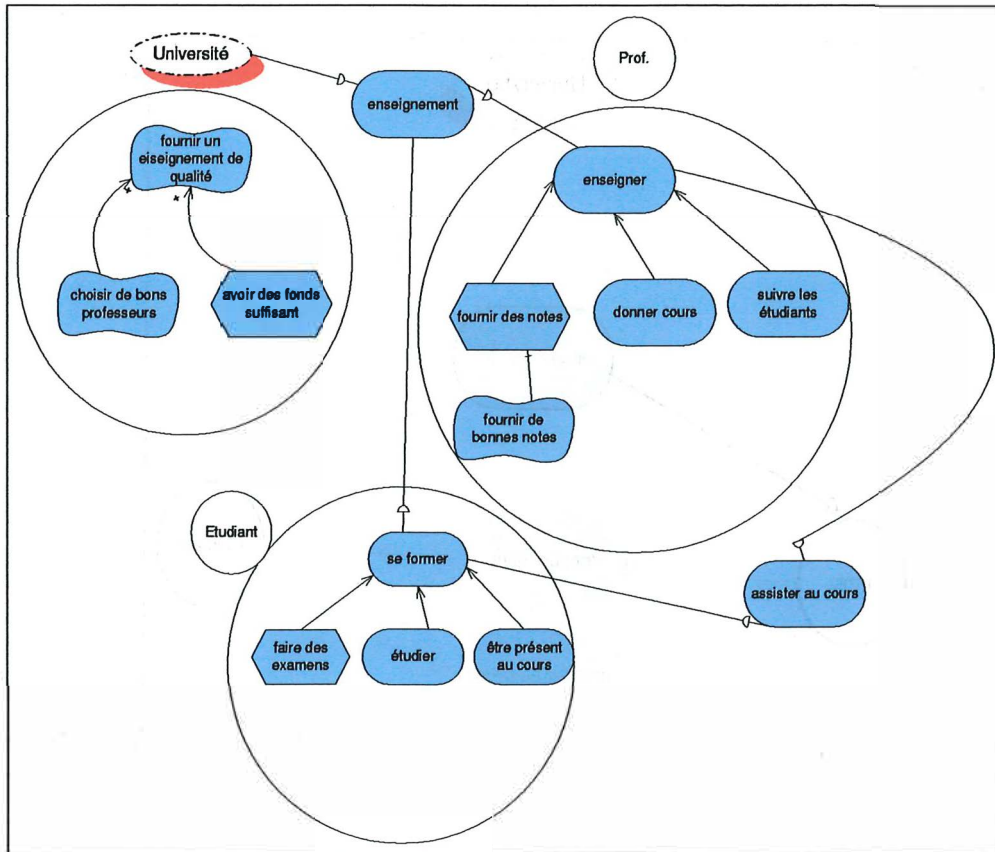
Voici ce que donnerait par exemple "le strategic dependency model" pour une université :

FIG. 6.3 – Strategic dependency model



et le "strategic rationale model" :

FIG. 6.4 – Strategic rationale model

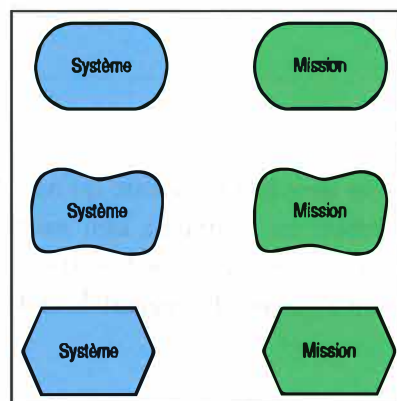


Cette am lioration est utile pour permettre d'inclure les buts poursuivis par les organisations mais n anmoins il peut nuire   la philosophie initiale du mod le i^*   savoir un concept orient  "acteur".

6.2.2 Buts de système et buts de mission

Comme on le sait, chaque acteur au sein d'une organisation est porteur d'un ensemble de buts de mission et de système qui lui sont propres. Il serait intéressant justement de faire la distinction entre les buts de système et les buts de mission. Le modèle i^* initial ne permet malheureusement pas de faire cette distinction. Une solution simple serait de changer la couleur des formes pour permettre de distinguer ces deux cas de figure.

FIG. 6.5 – Buts de système et buts de mission



Cette modification ne diminue pas la lisibilité des schémas et est très simple à mettre en oeuvre.

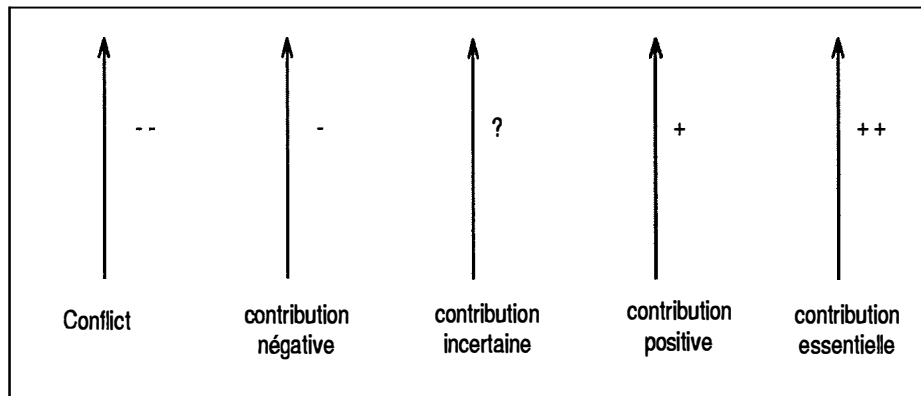
Grâce aux attributs, énoncés dans le chapitre précédent, il est tout à fait possible de caractériser davantage les buts de système en indiquant par un attribut s'il s'agit d'un but de survie, d'efficacité, de contrôle de l'environnement ou de croissance.

6.2.3 Les conflits entre buts

Mis à part pour le cas du "contribute to softgoal link" les conflits entre but dans le modèle i^* ne sont pas explicitement indiqués.

Un premier point à améliorer serait les "contribute to soft goal link". On pourrait les différencier davantage en reprenant ceux de l'approche *NFR*.

FIG. 6.6 – Contribute to soft-goal link



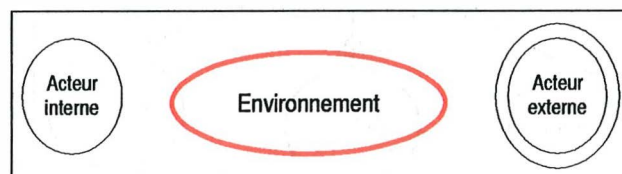
Cependant cette modification ne s'appliquerait qu'aux "soft goals" L'idéal pour les autres éléments serait de créer un nouveau lien entre les buts pouvant poser problème et cela indiquerait qu'il y a un conflit entre ces buts. Cependant cette solution assez simple à mettre en oeuvre chargerait lourdement les schémas.

6.3 Le pouvoir

6.3.1 Les relations avec l'environnement

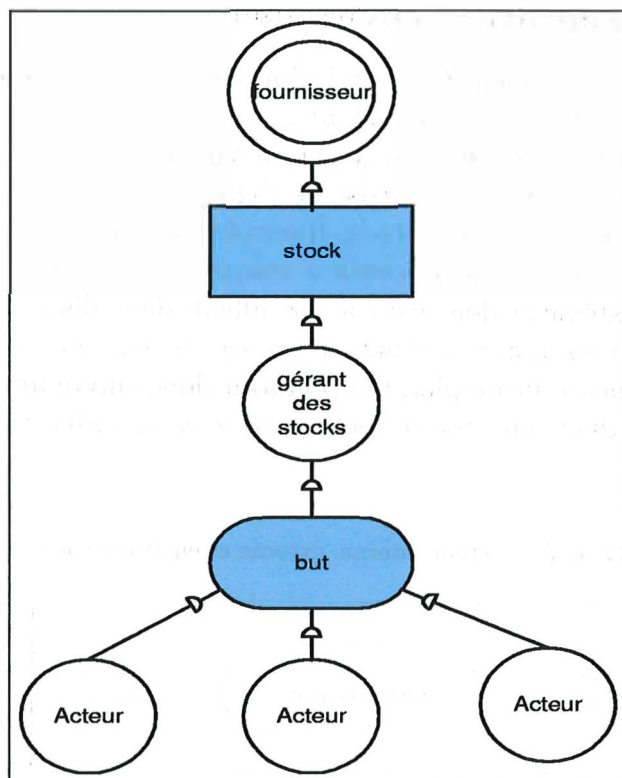
Nous avons vu précédemment que la relation avec l'environnement était difficile à modéliser dans le modèle i^* . Or pourtant c'est un concept assez important par le fait qu'aucune organisation ne peut survivre sans entretenir des relations avec l'environnement qui l'entoure. La maîtrise de l'environnement est donc comme on l'a vu au chapitre 2 une zone d'incertitude. Il serait donc intéressant de pouvoir bien la cerner. Cependant le modèle rend compte uniquement des relations entre les acteurs internes d'un système et donc c'est assez difficile de réaliser cela. Une façon de pallier à cela est d'utiliser la notion d'acteur externe développée au point précédent. Toutefois cela reste encore incomplet. On pourrait donc suivre un raisonnement similaire au point précédent concernant l'organisation en considérant l'environnement comme un acteur.

FIG. 6.7 – Acteur interne, externe et environnement



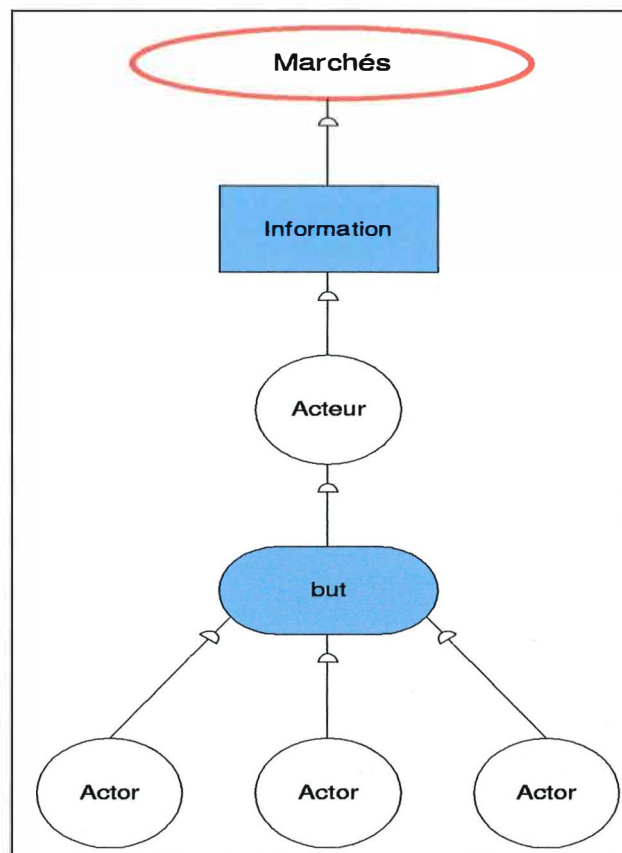
Pour illustrer ceci voici deux exemples. Dans le premier exemple on peut voir un gérant des stocks ayant des contacts avec l'extérieur :

FIG. 6.8 – Fournisseur externe et gérant des stocks



Dans le second un acteur ayant des connaissances des marchés, information qui peut être cruciale pour des objectifs poursuivis par des acteurs :

FIG. 6.9 – Acteur analysant le marché



Cette amélioration est assez pertinente du fait qu'elle permet de déceler une zone d'incertitude assez importante. Mais encore une fois l'introduction du concept environnement s'écarte de nouveau du modèle initial basé sur la notion d'acteur.

Conclusion

L'objectif principal de cette recherche est de montrer un formalisme pour modéliser, analyser et redessiner des processus. Ce formalisme se focalise principalement sur les types de processus trouvés dans la configuration de l'organisation où les gens, *les acteurs*, souvent intégrés dans des systèmes informatiques travaillent en collaboration. Le modèle i^* met en évidence et formalise une notion de processus qui est plus riche que dans les formalismes existants par le fait qu'il permet une plus grande ouverture dans l'analyse et la modélisation du comportement et les relations des agents dans un système. De plus le modèle i^* offre la possibilité de percevoir ces agents comme des acteurs sociaux avec des conflits d'intérêt et des motivations personnelles. En effet ces concepts, qui sont pourtant essentiels sont absents dans la plupart des modèles de conception classique.

Le modèle i^* consiste en deux composants, à savoir le "Strategic Dependency model" et le "Strategic Rationale model". Ces représentations graphiques riches ont permis de mettre en évidence les relations que les membres d'une organisation exercent entre eux à travers les concepts d'*acteur* ("depend" et "dependee"), *de type de dépendance*, *type du dépendum* ("goal", "softgoal", etc...), et de *degré de dépendance* (open, "committed" et "critical"). Ce modèle permet donc de détecter les conflits potentiels et permet de détecter si un acteur n'est pas capable d'accomplir un objectif à causes de plusieurs facteurs. Le modèle i^* apporte donc une approche orientée but (grâce aux liens "means-end") et orientée agent à l'analyse des exigences.

L'analyse organisationnelle est un processus fondamental dans la conception de système. Cependant il n'existe aucun outil permettant de modéliser de manière précise ces concepts essentiels. Le modèle i^* peut alors s'avérer très utile pour modéliser de manière formelle certains concepts clés des théories des organisations à savoir le concept d'acteur, de pouvoir ou de but spécifique et donc par conséquent intégrer ces éléments pertinents dans l'analyse des exigences de façon à permettre aux informaticiens de tenir compte de cela lors du développement du système d'information.

Le modèle i^* est aussi un modèle très flexible. Il permet d'adapter facilement des changements dans le système d'information sans pour autant faire table rase des modèles déjà conçus précédemment. En effet on peut aisément modifier le "Stratégic

dépendency model" ou le "Strategic Rationale model" pour un changement de rôle des acteurs.

Des outils puissants, performants et simples d'utilisation ont été développés afin de modéliser les schémas i^* sur ordinateur et de les intégrer à des documents. De plus ces outils peuvent facilement s'adapter à de nouvelles extensions du modèle i^* .

Bien que le modèle original souffre de quelques lacunes quant à la modélisation des concepts issus des théories des organisations, celles-ci peuvent être comblées par des extensions élémentaires du modèle existant.

Bibliographie

- [CF77] Michel Crozier and Erhard Friedberg. *L'acteur et le système*. Edition du Seuil, 1977.
- [CNM99] Lawrence Chung, Brian A. Nixon, and John Mylopoulos. *Non-Functional Requirements in software engineering*. Kluwer Academic Publishers, 1999.
- [Kav01] Evangelia Kavakli. Goal-oriented requirements engineering : A unifying framework. *RE*, 6(4) :237–251, November 2001.
- [Lam] Axel Van Lamsweerde. Grail/kaos : An environment for goal-driven requirements engineering. Université catholique de Louvain.
- [LM99] Claire Lobet-Maris. Analyse des organisations. Namur, 1999. Facultés Universitaires Notre-Dame de la Paix.
- [MFG] John Mylopoulos, Ariel Fuxmam, and Paolo Giorgini. From entities and relationships to social actors and dependencies. Departement of Computer Science, University of trento.
- [Min86] H. Mintzberg. *Le Pouvoir dans les organisations*. Edition d'Organisation/Agence d'Arc, Paris/Montreal, 1986.
- [Mor89] Gareth Morgan. *Images de l'organisation*. Les presses de l'université Laval, 1989.
- [NP95] Jean Nizet and François Pichault. *Comprendre les organisations, Mintzberg à l'épreuve des faits*. Gaëtan Morin éditeur, 1995.
- [Orl01] Wanda J. Orlikowski. Technology and institutions : What can research on information technology and research on organisations learn from each other? *Mis Quarterly*, 25(2) :145–165, Juin 2001.
- [Pfe81] J. Pfeffer. *Power in organisation*. Pitman Publishing, 1981.
- [Saa80] Thomas Saaty. *The Analytic Hierarchy Process*. McGraw-Hill Co., 1980.
- [YM94a] Eric Yu and John Mylopoulos. From e-r to "a-r" - modelling strategic actor relationships for buisness process reengineering. Departement of Computer Science, University of Toronto, December 1994.
- [YM94b] Eric Siu-Kwong Yu and John Mylopoulos. Towards modelling strategic actor relationships for information system development - with examples from buisness process reengineering. Departement of Computer Science, University of Toronto, October 1994.

- [Yu95] Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1995.
- [Yu01] Eric Yu. Agent orientation as modeling paradigm. *Wirtschaftsinformatik*, 43(2) :123–132, April 2001.

Annexe A

Le code de Redepend

Voici le code (en Visual Basic) rajouté à Redepend pour inclure la méthode AHP. Cela comprend 3 parties. D'une part des fonctions auxiliaires de base qui sont utilisées dans le code ahp, d'autre part le code qui permet de sélectionner les "goals" à inclure dans la matrice ainsi le code utilisé pour les fonctions qui permettent de bien placer les nombres sur les bons liens, et enfin le code de la feuille de calcul excel qui réalise les opérations de la méthode AHP proprement dite.

A.1 Fonctions auxiliaires

```
Option Explicit
Public Function MatrixToString(x() As String, ByVal n As Integer)
As String
Dim i, j As Integer
Dim y As String
On Error GoTo errorHandler
y = ""
For i = 0 To n - 1
For j = 0 To n - 1
If j = (n - 1) Then
y = CStr(y & x(i, j))
Else
y = CStr(y & x(i, j) & "n")
End If
Next
If i <> (n - 1) Then y = CStr(y & "r")
Next
MatrixToString = y & "r"
ExitHere:
Exit Function
errorHandler:
```

```
If Not dhError("MatrixOp:MatrixToString", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Function
Public Sub StringToMatrix(x() As String, ByVal y As String)
    Dim i, j, k As Integer
    Dim temp, temp2, tt As String
    On Error GoTo errorHandler
    i = 0
    j = 0
    temp2 = ""
    k = 0
    tt = y
    While k <> Len(tt)

        temp = Left(y, 1)
        y = Right(y, (Len(y) - 1))
        If temp = "n" Then
            x(i, j) = temp2
            j = j + 1
            temp2 = ""
            temp = ""
            k = k + 1
        ElseIf temp = "r" Then
            x(i, j) = temp2
            j = 0
            i = i + 1
            k = k + 1

            temp2 = ""
            temp = ""
        Else: temp2 = temp2 & temp
            temp = ""
            k = k + 1
        End If
    Wend
ExitHere:
Exit Sub
errorHandler:
```



```
If Not dhError("MatrixOp:StringToMatrix", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Sub

Public Function FindPosition(ByVal x As String, Data() As String,
    ByVal length As Integer) As Integer
    Dim i As Integer
    Dim StopLoop As Boolean
    StopLoop = False
    Dim position As Integer
    On Error GoTo errorHandler
    i = 0
    While (i <= length - 1) And Not (StopLoop)
        If Data(i) = x Then StopLoop = True
        i = i + 1
    Wend
    If StopLoop Then
        FindPosition = i - 1
    Else: FindPosition = 8
    End If
    ExitHere:
    Exit Function
errorHandler:

If Not dhError("MatrixOp:FindPosition", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Function

Public Sub build_data(data_in() As String, data_out() As String,
    ByVal length_in As Integer, ByVal length_out As Integer,
    NameOfExistingSG() As String, NameOfSG() As String, NewName() As String)
    Dim i, j, k, l As Integer
    Dim Name As String
    Dim position, pos As Integer
    Dim temp As String
    Dim StopLoop As Boolean

    On Error GoTo errorHandler
```

```

For i = 0 To length_out - 1
    NewName(i) = NameOfSG(i)
Next
Call Init_Data(data_out, length_out)
If SameArray(NameOfExistingSG, NameOfSG, length_in, length_out) Then
    'if there are no change ( no more subgoal,....)
    For i = 0 To length_out
        For j = 0 To length_out
            data_out(i, j) = data_in(i, j)
        Next
    Next
    Exit Sub
End If

If (length_out - length_in >= 0) Then
    'if 1 or more subgoal added
    For i = 0 To (length_out - 1)
        For j = 0 To (length_out - 1)
            NewName(j) = NameOfSG(j)
        Next
        Name = NameOfSG(i)
        position = FindPosition(Name, NameOfExistingSG, length_out)
        If (position <> 8) Then
            For j = 0 To length_out - 1
                pos = FindPosition(NameOfSG(j), NameOfExistingSG, length_out)
                If (pos <> 8) Then
                    data_out(i, j) = data_in(position, pos)
                End If
            Next
        End If
    Next
End If
Next
End If

If (length_out - length_in < 0) Then
    'if 1 or more subgoal removed
    For i = 0 To (length_out - 1)
        For j = 0 To (length_out - 1)
            NewName(j) = NameOfSG(j)
        Next
        Name = NameOfSG(i)
        position = FindPosition(Name, NameOfExistingSG, length_in)
        If (position <> 8) Then
            For j = 0 To length_out - 1

```

```

        pos = FindPosition(NameOfSG(j), NameOfExistingSG, length_in)
        If (pos <> 8) Then
            data_out(i, j) = data_in(position, pos)
        End If
    Next
End If
Next
For i = 0 To length_out - 1
    If EmptyLine(data_out, length_out, i) Then
        j = i + 1
        StopLoop = False
        While ((j <= length_out - 1) And (StopLoop = False))
            If EmptyLine(data_out, length_out, j) Then
                StopLoop = True
            Else: j = j + 1
            End If
        Wend
        If (StopLoop = True) Then
            For k = i + 1 To (j - 1)
                For l = 0 To length_out
                    data_out(k, l) = data_out(k + 1, l)
                Next
                NewName(k) = NameOfExistingSG(k + 1)
            Next
            NewName(j - 1) = NameOfExistingSG(i)
        End If
    End If
Next
End If
Call Init_diag(data_out, length_out)
ExitHere:
Exit Sub
errorHandler:

If Not dhError("MatrixOp:build_data", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Sub
Public Sub Init_diag(Data() As String, ByVal length As Integer)
    'pre
    'post the diagonal of data is 1

```

```

Dim i, j As Integer
On Error GoTo errorHandler
For i = 0 To length - 1
    Data(i, i) = "1"
Next
ExitHere:
Exit Sub
errorHandler:

If Not dhError("MatrixOp:Init_Diag", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Sub

Public Sub Init_Data(Data() As String, ByVal length As Integer)
    'post all value of data is null
    Dim i, j As Integer
    On Error GoTo errorHandler
    For i = 0 To length - 1
        For j = 0 To length - 1
            Data(i, j) = ""
        Next
    Next
ExitHere:
Exit Sub
errorHandler:

If Not dhError("MatrixOp:Init_data", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Sub

Public Sub build_result(Result() As Double, Name() As String,
    NewName() As String, ByVal length As Integer)
    'post result contains the ahp result corresponding with the
    subgoal name (address of ahp result and name of subgoal is the same)
    Dim temp(7) As Double
    Dim i, pos As Integer
    On Error GoTo errorHandler
    For i = 0 To length - 1
        pos = FindPosition(NewName(i), Name, length)
    
```

```
temp(pos) = Result(i)
Next
For i = 0 To length - 1
Result(i) = temp(i)
Next
ExitHere:
Exit Sub
errorHandler:

If Not dhError("MatrixOp:build_result", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Sub

Public Function SameArray(x() As String, y() As String, ByVal lx As Integer,
ByVal ly As Integer) As Boolean
    'post SameArray = (x=y)
    Dim i As Integer
    Dim StopLoop As Boolean
    On Error GoTo errorHandler
    If (lx = ly) Then
        i = 0
        StopLoop = False
        While (i <= lx - 1) And Not (StopLoop)
            If (x(i) <> y(i)) Then
                StopLoop = True
            Else: i = i + 1
        End If
    Wend
    If StopLoop = True Then
        SameArray = False
    Else: SameArray = True
    End If
    Else: SameArray = False
    End If
ExitHere:
Exit Function
errorHandler:

If Not dhError("MatrixOp:SameArray", False) Then
    'Cancel has been clicked
    'do something for the error if required
```

```

End If
Resume ExitHere
End Function
Public Function EmptyLine(x() As String, ByVal length As Integer,
ByVal l As Integer) As Boolean
'EmptyLine = (the line l of x is empty)
Dim i As Integer
Dim StopLoop As Boolean
On Error GoTo errorHandler
i = 0
StopLoop = False
While i <= length - 1 And Not StopLoop
If x(l, i) = "" Then
i = i + 1
Else: StopLoop = True
End If
Wend
If StopLoop Then
EmptyLine = False
Else: EmptyLine = True
End If
ExitHere:
Exit Function
errorHandler:

If Not dhError("MatrixOp:EmptyLine", False) Then
'Cancel has been clicked
'do something for the error if required
End If
Resume ExitHere
End Function
Public Function EmptyArray(x() As Double, ByVal length As Integer)
As Boolean
'emptyarray = (all value of x are 1 or 0)
Dim i As Integer
On Error GoTo errorHandler
EmptyArray = True
While i <= length - 1 And EmptyArray
If Not (x(i) = 1 Or x(i) = 0) Then EmptyArray = False
i = i + 1
Wend
ExitHere:
Exit Function

```

```
errorHandler:
```

```
If Not dhError("MatrixOp:EmptyArray", False) Then
    'Cancel has been clicked
    'do something for the error if required
End If
Resume ExitHere
End Function
```

A.2 Le code AHP du template visio

```
Option Explicit
Private sel As Visio.Selection
Private Connection As Visio.Connect
Private NbrAlternatives As Integer
Private strNodeType As String
Private strNodeText As String
Private LineExcel(7) As String
Private ColumnExcel(7) As String
Private NameOfDataCell(7, 7) As String
Private ColumnOfResult(7) As String

Public Sub ListOfAlternatives(x() As String)

    Dim y, yy As Visio.Connect
    Dim NameOfGoal, temp As String
    Dim pag As Visio.Page
    Dim IdLink
    Dim i As Integer
    Dim shap As Visio.shape
    On Error GoTo errorHandler
    With sel.Item(1)
        i = 0
        NameOfGoal = sel.Item(1).Text
        For Each y In .FromConnects
            IdLink = y.FromSheet.ID
            Set pag = ActivePage
            Set shap = pag.Shapes.ItemFromID(IdLink)
            For Each yy In shap.connects
                If (RemoveDoubleQuotes(yy.FromSheet.Cells
                    ("User.AHP").Formula) = "") Then
                    If Not (yy.ToSheet.Text = NameOfGoal) Then
```

```

        yy.FromSheet.Text = ""
        x(i) = yy.ToSheet.Text
        i = i + 1

    End If
End If
Next
Next
End With
NbrAlternatives = i
ExitHere:

Exit Sub
errorHandler:

    If Not dhError("clsAhp:ListOfAlternatives", False) Then
        'Cancel has been clicked
        'do something for the error if required
    End If
    Resume ExitHere
End Sub

Public Sub AhpLevel3()

Dim temp, ValueOfAttribute, ValueOfAlternative, Alternative,
    TempShap As String
Dim Result As Double
Dim NameOfGoal, nam As String
Dim IdLink, IdLink2, id1
Dim j As Integer
Dim ArrayOfAlternatives(7) As String
Dim pag As Visio.Page
Dim shap, shapB, shapC As Visio.shape
Dim y, LinkConnected, yyy, yyyy, z, zz, zzz As Visio.Connect
On Error GoTo errorHandler
Set sel = ActiveWindow.Selection
NbrAlternatives = 0
Result = 0
Call ListOfAlternatives(ArrayOfAlternatives)
If NbrAlternatives = 0 Then
MsgBox "There are no Alternatives Connected", vbCritical, "Ahp"

```



```

Set shapC = pag.Shapes.ItemFromID(id1)
For Each zzz In shapC.connects
    If Not (zzz.ToSheet.Text = nam) Then
        If (zzz.ToSheet.Text = temp) Then
            ValueOfAlternative = RemoveDoubleQuotes
            (zzz.FromSheet.Cells("User.ahp").Formula)
            If CDb1(ValueOfAlternative) = 0 Then
                MsgBox "The diagram is incomplete",
                    vbCritical, "AHP"
                Exit Sub
            End If
            If CDb1(ValueOfAttribute) = 0 Then
                MsgBox "The diagram is incomplete",
                    vbCritical, "AHP"
                Exit Sub
            End If

            Result = Result + (CDbl(ValueOfAlternative) *
                CDb1(ValueOfAttribute))

                GoTo 11
            End If
        End If
    Next
Next
End If

    End If
Next
Next
End With
End If
End If
11:    Next

        Next
End With

With sel.Item(1)
    NameOfGoal = sel.Item(1).Text
    For Each y In .FromConnects

```

```

        IdLink = y.FromSheet.ID
        Set pag = ActivePage
        Set shap = pag.Shapes.ItemFromID(IdLink)
        For Each LinkConnected In shap.connects
            'MsgBox LinkConnected.ToSheet.Text
            TempShap = RemoveDoubleQuotes
            (LinkConnected.FromSheet.Cells("User.Shape").Formula)
                If (TempShap = "Means-ends") Then

                If Not (LinkConnected.ToSheet.Text = NameOfGoal) Then
                    If LinkConnected.ToSheet.Text = Alternative Then
                        If Result = 0 Then
                            MsgBox "The diagram is incomplete", vbCritical, "AHP"
                            Exit Sub
                        End If
                        LinkConnected.FromSheet.Text = Result

                    End If
                End If
            End If
        Next
    Next
End With

Next
ExitHere:

Exit Sub
errorHandler:

    If Not dhError("clsAhp:AhpLevel3", False) Then
        'Cancel has been clicked
        'do something for the error if required
    End If
    Resume ExitHere
End Sub

Public Sub AhpLevel2()

    Dim Result(7) As Double
    Dim StrResult As String
    Dim NameOfSubGoal(7) As String

```

```

Dim NumberOfSubGoal, nbr As Integer
Dim Data(7, 7) As String
Dim data2(7, 7) As String

Dim NewName(7) As String
Dim NameOfShape, temp, NameOfLink As String
Dim NameOfGoal As String
Dim pag As Visio.Page
Dim objXL As Excel.Application
Dim objWkb As Excel.Workbook
Dim objSht As Excel.Worksheet
Dim y, LinkConnected As Visio.Connect
Dim i, j, pos As Integer
Dim shap As Visio.shape
Dim k, strgoal As String
Dim IdLink
Dim Extention As String
'Dim SubGoal(7) As String
Dim NameOfExistingSG(7) As String
Dim Name As String
Const conSHT_NAME = "AHP"
Const conWKB_NAME = "C:\ahp\ahp.xls"
Dim shpObj As Visio.shape

On Error GoTo errorHandler

SubGoal(0) = "User.SubGoal_1"
SubGoal(1) = "User.SubGoal_2"
SubGoal(2) = "User.SubGoal_3"
SubGoal(3) = "User.SubGoal_4"
SubGoal(4) = "User.SubGoal_5"
SubGoal(5) = "User.SubGoal_6"
SubGoal(6) = "User.SubGoal_7"
'begin initialize the path where the AHP excel spreadsheet is.
Path = ActivePage.PageSheet.Cells(PathExcel).Formula
Path = RemoveDoubleQuotes(Path)
If Path = "" Then
    Path = conWKB_NAME
Else:
    Extention = Right(Path, 4)
    If StrComp(".xls", Extention, vbTextCompare) <> 0 Then
        Path = Path & "\Ahp.xls"
    
```

```
End If
End If
'end initialize path
' initialisation of arrays witch contains the address
of interesting cells of the spreadsheet
LineExcel(0) = "B1"
LineExcel(1) = "C1"
LineExcel(2) = "D1"
LineExcel(3) = "E1"
LineExcel(4) = "F1"
LineExcel(5) = "G1"
LineExcel(6) = "H1"
ColumnExcel(0) = "A2"
ColumnExcel(1) = "A3"
ColumnExcel(2) = "A4"
ColumnExcel(3) = "A5"
ColumnExcel(4) = "A6"
ColumnExcel(5) = "A7"
ColumnExcel(6) = "A8"
ColumnOfResult(0) = "K2"
ColumnOfResult(1) = "K3"
ColumnOfResult(2) = "K4"
ColumnOfResult(3) = "K5"
ColumnOfResult(4) = "K6"
ColumnOfResult(5) = "K7"
ColumnOfResult(6) = "K8"
NumberOfSubGoal = 0
NameOfDataCell(0, 0) = "B2"
NameOfDataCell(0, 1) = "C2"
NameOfDataCell(0, 2) = "D2"
NameOfDataCell(0, 3) = "E2"
NameOfDataCell(0, 4) = "F2"
NameOfDataCell(0, 5) = "G2"
NameOfDataCell(0, 6) = "H2"
NameOfDataCell(1, 0) = "B3"
NameOfDataCell(1, 1) = "C3"
NameOfDataCell(1, 2) = "D3"
NameOfDataCell(1, 3) = "E3"
NameOfDataCell(1, 4) = "F3"
NameOfDataCell(1, 5) = "G3"
NameOfDataCell(1, 6) = "H3"
NameOfDataCell(2, 0) = "B4"
NameOfDataCell(2, 1) = "C4"
```

```

NameOfDataCell(2, 2) = "D4"
NameOfDataCell(2, 3) = "E4"
NameOfDataCell(2, 4) = "F4"
NameOfDataCell(2, 5) = "G4"
NameOfDataCell(2, 6) = "H4"
NameOfDataCell(3, 0) = "B5"
NameOfDataCell(3, 1) = "C5"
NameOfDataCell(3, 2) = "D5"
NameOfDataCell(3, 3) = "E5"
NameOfDataCell(3, 4) = "F5"
NameOfDataCell(3, 5) = "G5"
NameOfDataCell(3, 6) = "H5"
NameOfDataCell(4, 0) = "B6"
NameOfDataCell(4, 1) = "C6"
NameOfDataCell(4, 2) = "D6"
NameOfDataCell(4, 3) = "E6"
NameOfDataCell(4, 4) = "F6"
NameOfDataCell(4, 5) = "G6"
NameOfDataCell(4, 6) = "H6"
NameOfDataCell(5, 0) = "B7"
NameOfDataCell(5, 1) = "C7"
NameOfDataCell(5, 2) = "D7"
NameOfDataCell(5, 3) = "E7"
NameOfDataCell(5, 4) = "F7"
NameOfDataCell(5, 5) = "G7"
NameOfDataCell(5, 6) = "H7"
NameOfDataCell(6, 0) = "B8"
NameOfDataCell(6, 1) = "C8"
NameOfDataCell(6, 2) = "D8"
NameOfDataCell(6, 3) = "E8"
NameOfDataCell(6, 4) = "F8"
NameOfDataCell(6, 5) = "G8"
NameOfDataCell(6, 6) = "H8"
'end initialisation
'begin loop each connection of the selected shape to find the name
of the subgoals
Set sel = ActiveWindow.Selection
  With sel.Item(1)
    NameOfGoal = sel.Item(1).Text
    For Each y In .FromConnects
      IdLink = y.FromSheet.ID
      Set pag = ActivePage
      Set shap = pag.Shapes.ItemFromID(IdLink)

```

```

For Each LinkConnected In shap.connects
    If Not (LinkConnected.ToSheet.Text = NameOfGoal) Then
        NameOfShape = RemoveDoubleQuotes
        (LinkConnected.ToSheet.Cells("User.Shape").Formula)
        NameOfLink = RemoveDoubleQuotes
        (LinkConnected.FromSheet.Cells("User.Shape").Formula)
        If Not (LinkConnected.FromPart =
            visBegin And NameOfShape = "Goal") Then
            If Not (LinkConnected.FromPart =
                visBegin And NameOfShape = "Soft Goal") Then
                If Not (NameOfLink = "Dependency") Then
                    If Not (NameOfLink = "Task Decomposition") Then
                        If Not (NameOfShape = "Resource") Then
                            If Not (NameOfLink = "AdditionalDepLink")
                                Then

                                NumberOfSubGoal = NumberOfSubGoal + 1
                                If NumberOfSubGoal <= 7 Then
                                    NameOfSubGoal(NumberOfSubGoal - 1)
                                    = LinkConnected.ToSheet.Text
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
Next

Next
End With

If NumberOfSubGoal <= 1 Or NumberOfSubGoal > 7 Then
    MsgBox " AHP Cannot run because you need a minimum of 2 sub-goal
    and a maximum of 7 subgoal", vbOKOnly + vbCritical, ("AHP")
    Exit Sub
End If

'end loop each connection of the selected shape
'lauch excel and add the name of the subgoal into it and when it's
possible put also the values
Set objXL = New Excel.Application

```

With objXL

```

' objXL.DisplayClipboardWindow = False
Set objWkb = .Workbooks.Open(Path)
Set objSht = objWkb.Worksheets(conSHT_NAME)

.Range("E11").value = NumberOfSubGoal
.Range("K11").value = 0

With sel.Item(1)
    If Not (RemoveDoubleQuotes(sel.Item(1).Cells
    ("User.size").Formula) = "") Then
        'if there are a previous calculation of AHP
        nbr = CInt(RemoveDoubleQuotes(sel.Item(1).Cells
        ("User.size").Formula))
        strgoal = RemoveDoubleQuotes(sel.Item(1).Cells
        ("User.AHP_DATA").Formula)
        Call StringToMatrix(Data, strgoal)
        For i = 0 To nbr
            NameOfExistingSG(i) = RemoveDoubleQuotes
            (sel.Item(1).Cells(SubGoal(i)).Formula)
        Next
    End If
End With
' if there are a previous calculation put the previous values
If Not (RemoveDoubleQuotes(sel.Item(1).Cells("User.size").Formula) = "")
Then
    Call build_data(Data, data2, nbr, NumberOfSubGoal, NameOfExistingSG,
    NameOfSubGoal, NewName)
    For i = 0 To NumberOfSubGoal - 1
        .Range(ColumnExcel(i)).value = NewName(i)
        .Range(LineExcel(i)).value = NewName(i)
    Next
    'put the values
    For j = 0 To NumberOfSubGoal - 1
        For i = 0 To NumberOfSubGoal - 1
            If Not (data2(i, j) = "") Then
                .Range(NameOfDataCell(i, j)).value = CDBl(data2(i, j))
            End If
        Next
    Next
Else 'if there are no previous calculation , simply put the

```



```

        names of the subgoal
        For i = 0 To NumberOfSubGoal - 1
            .Range(ColumnExcel(i)).value = NameOfSubGoal(i)
            .Range(LineExcel(i)).value = NameOfSubGoal(i)
        Next
    End If
    .Visible = True
waiting:

If (.Range("K11").value = 0) Then GoTo waiting
' this loop is for waiting the end of Excel
'saving data...

For i = 0 To NumberOfSubGoal - 1
    Result(i) = .Range(ColumnOfResult(i)).value
Next
If Not (RemoveDoubleQuotes(sel.Item(1).Cells
("User.size").Formula) = "") Then
    Call build_result(Result, NameOfSubGoal, NewName,
        NumberOfSubGoal)
End If
For i = 0 To NumberOfSubGoal
    For j = 0 To NumberOfSubGoal
        data2(i, j) = .Range(NameOfDataCell(i, j)).value
    Next
Next
'end saving data
'closing excel
objWkb.Saved = True
objWkb.Close
objXL.Visible = False
objXL.Quit
Set objSht = Nothing
Set objWkb = Nothing
Set objXL = Nothing

End With
i = 0
With sel.Item(1)
    If EmptyArray(Result, NumberOfSubGoal) Then
        .Cells("User.AHP_DATA").Formula = "=" & Chr(34) & Chr(34)
        .Cells("User.size").Formula = "=" & Chr(34) & Chr(34)
        For j = 0 To 6

```

```

        .Cells(SubGoal(j)).Formula = "=" & Chr(34) & Chr(34)
    Next
Else
    temp = MatrixToString(data2, NumberOfSubGoal)
    .Cells("User.AHP_DATA").Formula = "=" & Chr(34) & temp &
    Chr(34)
    .Cells("User.size").Formula = "=" & Chr(34) &
    CStr(NumberOfSubGoal) & Chr(34)
    For j = 0 To NumberOfSubGoal - 1
        .Cells(SubGoal(j)).Formula = "=" & Chr(34) &
        NameOfSubGoal(j) & Chr(34)
    Next
    For j = NumberOfSubGoal To 6
        .Cells(SubGoal(j)).Formula = "=" & Chr(34) & Chr(34)
    Next
End If
'put the value on the arrows
NameOfGoal = sel.Item(1).Text
For Each y In .FromConnects
    IdLink = y.FromSheet.ID
    Set pag = ActivePage
    Set shap = pag.Shapes.ItemFromID(IdLink)
    For Each LinkConnected In shap.connects
        If Not (LinkConnected.ToSheet.Text = NameOfGoal) Then
            NameOfShape = RemoveDoubleQuotes(LinkConnected.ToSheet.Cells
            ("User.Shape").Formula)
            NameOfLink = RemoveDoubleQuotes(LinkConnected.FromSheet.Cells
            ("User.Shape").Formula)
            If Not (LinkConnected.FromPart = visBegin And NameOfShape
            = "Goal") Then
                If Not (LinkConnected.FromPart = visBegin And NameOfShape
                = "Soft Goal") Then
                    If Not (NameOfLink = "Dependency") Then
                        If Not (NameOfLink = "Task Decomposition") Then
                            If Not (NameOfShape = "Resource") Then
                                If Not (NameOfLink = "AdditionalDepLink") Then
                                    If Result(i) = 1 Or Result(i) = 0 Then
                                        shap.Text = ""
                                        shap.Cells("User.Ahp").Formula = "=" & Chr(34)
                                        & Chr(34)
                                        LinkConnected.ToSheet.Cells("User.Ahp")
                                        .Formula = "=" & Chr(34) & Chr(34)
                                        i = i + 1
                                    End If
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    Next
Next

```

```

Else
    StrResult = CStr(Result(i))
    shap.Text = StrResult
    shap.Cells("User.Ahp").Formula = "=" &
    Chr(34) & StrResult & Chr(34)
    LinkConnected.ToSheet.Cells("User.Ahp")
    .Formula = "=" & Chr(34) & StrResult &
    Chr(34)
    i = i + 1

End If
End If
End If
End If
End If
End If
End With
ExitHere:

```

```

Exit Sub
errorHandler:
    Set objSht = Nothing
    Set objWkb = Nothing
    Set objXL = Nothing
    If Not dhError("clsAhp:AhpLevel2", False) Then
        'Cancel has been clicked
        'do something for the error if required
    End If
    Resume ExitHere
End Sub

```

```

Private Sub Class_Initialize()
On Error GoTo errorHandler
Set sel = ActiveWindow.Selection
ExitHere:
Exit Sub
errorHandler:
    If Not dhError("clsAhp:Class_Initialize", False) Then

```

```

        'Cancel has been clicked
        'do something for the error if required
    End If
    Resume ExitHere
End Sub

Private Sub Class_Terminate()
On Error GoTo errorHandler
Set sel = Nothing
ExitHere:
Exit Sub
errorHandler:
    If Not dhError("clsAhp:Class_Terminate", False) Then
        'Cancel has been clicked
        'do something for the error if required
    End If
    Resume ExitHere
End Sub

Public Sub ViewValue()
Dim temp As String
With sel.Item(1)
temp = RemoveDoubleQuotes(.Cells(cntAHPCell).Formula)
If temp = "" Then
MsgBox "There is no value. Please lauch AHP", vbInformation, "AHP"
Else
AHPBox.TextBox1 = temp
AHPBox.Show
End If
End With
End Sub

```

A.3 Le code AHP de la feuille de calcul Excel

Option Explicit

```

Private Sub CommandButton1_Click()
Dim tabl
Dim unfilled As Boolean

Dim I As Integer, J As Integer
Size = Range("E11").Value
If Size <> 0 Then
tabl = Range("B2:H8").Value

```

```
unfilled = True
For I = 1 To Size
    For J = 1 To Size
        If Not ((tabl(I, J)) = 0) Then
            If Not (tabl(I, J)) = 1 Then
                unfilled = False
            End If
        End If
    Next J
Next I
If Not unfilled Then

    For I = 1 To Size
        For J = 1 To Size
            If (tabl(I, J)) = 0 Then
                MsgBox "All of the cells need to be filled in the matrix",
                vbCritical, "AHP"
            End If
        Next J
    Next I
End If
Range("K11").Value = 1
End Sub

Private Sub CommandButton2_Click()
    'this procedure is for the wizard button

    Dim I As Integer
    Dim Ligne1(6) As String
    Ligne1(0) = "C2"
    Ligne1(1) = "D2"
    Ligne1(2) = "E2"
    Ligne1(3) = "F2"
    Ligne1(4) = "G2"
    Ligne1(5) = "H2"

    Dim Ligne2(5) As String
    Ligne2(0) = "D3"
```

```
Ligne2(1) = "E3"
Ligne2(2) = "F3"
Ligne2(3) = "G3"
Ligne2(4) = "H3"
```

```
Dim Ligne3(4) As String
Ligne3(0) = "E4"
Ligne3(1) = "F4"
Ligne3(2) = "G4"
Ligne3(3) = "H4"
```

```
Dim Ligne4(3) As String
Ligne4(0) = "F5"
Ligne4(1) = "G5"
Ligne4(2) = "H5"
```

```
Dim Ligne5(2)
Ligne5(0) = "G6"
Ligne5(1) = "H6"
```

```
Dim Ligne6(1)
Ligne6(0) = "H7"
'just to avois some problems
Range("H9").Select
If Range("E11").Value = 0 Then
GoTo AskSize
Else: Size = Range("E11").Value
GoTo debut
End If
AskSize: Size = InputBox("What's the matrix's size ? Please enter a
value between 2 and 7")
If Not ((Size = "2") Or (Size = "3") Or (Size = "4") Or (Size = "5")
Or (Size = "6") Or (Size = "7")) Then
MsgBox ("Size is not correct, try again..."), vbOKOnly + vbCritical,
("AHP")
GoTo AskSize
End If
debut: CancelWizard = False
For I = 0 To (Size - 2)
If Not (CancelWizard) Then
Range(Ligne1(I)).Select
Else: Exit Sub
End If
```

```
Next
If (Size > 2) Then
For I = 0 To (Size - 3)
If Not (CancelWizard) Then
Range(Ligne2(I)).Select
Else: Exit Sub
End If
Next
Else: Exit Sub
End If
If (Size > 3) Then
For I = 0 To (Size - 4)
If Not (CancelWizard) Then
Range(Ligne3(I)).Select
Else: Exit Sub
End If
Next
Else:
Exit Sub
End If
If (Size > 4) Then
For I = 0 To (Size - 5)
If Not (CancelWizard) Then
Range(Ligne4(I)).Select
Else: Exit Sub
End If
Next
Else: Exit Sub
End If
If (Size > 5) Then
For I = 0 To (Size - 6)
If Not (CancelWizard) Then
Range(Ligne5(I)).Select
Else: Exit Sub
End If
Next
Else: Exit Sub
End If
If (Size > 6) Then
For I = 0 To (Size - 7)
If Not (CancelWizard) Then
Range(Ligne6(I)).Select
Else: Exit Sub
```

```
End If
Next
Else: Exit Sub
End If
```

```
End Sub
```

```
Private Sub Worksheet_Change(ByVal Target As Excel.Range)
'if there is a change of a cell of the matrix...
```

```
If (Not Intersect(Target, Range("B2")) Is Nothing) Or
(Not Intersect(Target, Range("C3")) Is Nothing) Or
(Not Intersect(Target, Range("D4")) Is Nothing) Or
(Not Intersect(Target, Range("E5")) Is Nothing) Or
(Not Intersect(Target, Range("F6")) Is Nothing) Or
(Not Intersect(Target, Range("G7")) Is Nothing) Or
(Not Intersect(Target, Range("H8")) Is Nothing) Then
'if there is a change of one cell of the matrix then
' if the cell is on the diagonal, then the macro checks if it's 1
```

```
    If Not (Range("B2").Value = 1) Then
        MsgBox ("B2 must be 1"), vbOKOnly + vbCritical, ("AHP")
        Range("B2").Value = "1"
```

```
    Exit Sub
End If
    If Not (Range("C3").Value = 1) Then
        MsgBox ("C3 must be 1"), vbOKOnly + vbCritical, ("AHP")
        Range("C3").Value = "1"
    Exit Sub
End If
    If Not (Range("D4").Value = 1) Then
        MsgBox ("D4 must be 1"), vbOKOnly + vbCritical, ("AHP")
        Range("D4").Value = "1"
    Exit Sub
```

```
End If
    If Not (Range("E5").Value = 1) Then
        MsgBox ("E5 must be 1"), vbOKOnly + vbCritical, ("AHP")
```



```

    Range("E5").Value = "1"
    Exit Sub
End If
If Not (Range("F6").Value = 1) Then
    MsgBox ("F6 must be 1"), vbOKOnly + vbCritical, ("AHP")
    Range("F6").Value = "1"
    Exit Sub
End If
If Not (Range("G7").Value = 1) Then
    MsgBox ("G7 must be 1"), vbOKOnly + vbCritical, ("AHP")
    Range("G7").Value = "1"
    Exit Sub
End If
If Not (Range("H8").Value = 1) Then
    MsgBox ("H8 must be 1"), vbOKOnly + vbCritical, ("AHP")
    Range("H8").Value = "1"
    Exit Sub
End If
Else: Exit Sub
End If
End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    'if a cell is selected
    'pre
    'post the cell Aij contains a Saaty's scale value
    '    and the cell Aji contains the opposite value of the cell Aij
    Dim AddrOpp
    Dim temp As String
    Dim Colonne As String
    Dim Ligne As String
    Dim Name1 As String
    Dim Name2 As String
    Dim x, y, z As Double
    If (Not Intersect(Target, Range("B2:H8"))) Is Nothing) Then
        Call Colligne(Colonne, Ligne)
        Adresse = Colonne & Ligne
        Name1 = Colonne & "1"
        Name2 = "A" & Ligne
        temp = Range(Name2).Value & " in relation to " & Range(Name1).Value
        UserForm1.TextBox1.Text = temp
        Load UserForm1
        UserForm1.Show
    End If
End Sub

```

```

Call FindOpposite(Adresse, AddrOpp)
If Not (Range(Adresse).Value = 0) And Not (AddrOpp = "") Then
    z = Application.WorksheetFunction.round(1 / Range(Adresse).Value, 2)
    y = Range(Adresse).Value
    x = Range(Adresse).Application.WorksheetFunction.round(y, 2)
    Range(Adresse).Value = x
    Range(AddrOpp).Value = z
Else: If Not (AddrOpp = "") Then Range(AddrOpp).Value = ""
End If
End If
End Sub
Sub Colligne(Colonne, Ligne)

'pre: a cell is active
'post: colonne is the column of the active cell
'     ligne is the line of the active cell

Colonne = Left$(ActiveCell.Address(0, 0), (ActiveCell.Column < 27) + 2)
Ligne = ActiveCell.Row

End Sub
Sub FindOpposite(ByVal Adresse, AddrOpp)

'pre : Adresse is the address of a cell Aij in the matrix
'post : Adresse don't change
'     AddrOpp is the adresse of the cell Aji in the matrix

Dim Ligne1(6) As String
Ligne1(0) = "C2"
Ligne1(1) = "D2"
Ligne1(2) = "E2"
Ligne1(3) = "F2"
Ligne1(4) = "G2"
Ligne1(5) = "H2"
Dim Colonne1(6) As String
Colonne1(0) = "B3"
Colonne1(1) = "B4"
Colonne1(2) = "B5"
Colonne1(3) = "B6"
Colonne1(4) = "B7"
Colonne1(5) = "B8"
Dim Ligne2(5) As String
Ligne2(0) = "D3"

```

```
Ligne2(1) = "E3"
Ligne2(2) = "F3"
Ligne2(3) = "G3"
Ligne2(4) = "H3"
Dim Colonne2(5) As String
Colonne2(0) = "C4"
Colonne2(1) = "C5"
Colonne2(2) = "C6"
Colonne2(3) = "C7"
Colonne2(4) = "C8"
Dim Ligne3(4) As String
Ligne3(0) = "E4"
Ligne3(1) = "F4"
Ligne3(2) = "G4"
Ligne3(3) = "H4"
Dim Colonne3(4) As String
Colonne3(0) = "D5"
Colonne3(1) = "D6"
Colonne3(2) = "D7"
Colonne3(3) = "D8"
Dim Ligne4(3) As String
Ligne4(0) = "F5"
Ligne4(1) = "G5"
Ligne4(2) = "H5"
Dim Colonne4(3) As String
Colonne4(0) = "E6"
Colonne4(1) = "E7"
Colonne4(2) = "E8"
Dim Ligne5(2)
Ligne5(0) = "G6"
Ligne5(1) = "H6"
Dim Colonne5(2)
Colonne5(0) = "F7"
Colonne5(1) = "F8"
Dim Ligne6(1)
Ligne6(0) = "H7"
Dim Colonne6(1)
Colonne6(0) = "G8"
Dim Halt As Boolean
Dim I As Integer
I = 0
While (I <= 5) And (Not Halt)
If Ligne1(I) = Adresse Then
```

```
AddrOpp = Colonne1(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 4) And (Not Halt)
If Ligne2(I) = Adresse Then
AddrOpp = Colonne2(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 3) And (Not Halt)
If Ligne3(I) = Adresse Then
AddrOpp = Colonne3(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 2) And (Not Halt)
If Ligne4(I) = Adresse Then
AddrOpp = Colonne4(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 1) And (Not Halt)
If Ligne5(I) = Adresse Then
AddrOpp = Colonne5(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
```

```
While (I <= 0) And (Not Halt)
If Ligne6(I) = Adresse Then
AddrOpp = Colonne6(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 5) And (Not Halt)
If Colonne1(I) = Adresse Then
AddrOpp = Ligne1(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 4) And (Not Halt)
If Colonne2(I) = Adresse Then
AddrOpp = Ligne2(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 3) And (Not Halt)
If Colonne3(I) = Adresse Then
AddrOpp = Ligne3(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 2) And (Not Halt)
If Colonne4(I) = Adresse Then
AddrOpp = Ligne4(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
```

```

Wend
I = 0
While (I <= 1) And (Not Halt)
If Colonne5(I) = Adresse Then
AddrOpp = Ligne5(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
I = 0
While (I <= 0) And (Not Halt)
If Colonne6(I) = Adresse Then
AddrOpp = Ligne6(I)
Halt = True
Exit Sub
Else: I = I + 1
End If
Wend
End Sub
Public Adresse As String
Public Size
Public CancelWizard As Boolean

Dim aRemove(84) As Integer
Dim iRemove As Integer
Public Sub Auto_Open()
    With Application
        .DisplayFormulaBar = False
        .DisplayStatusBar = False
        .WindowState = xlNormal
        .Height = 220
        .Width = 700

        .Caption = "Ahp"
    For iRemove = 1 To Application.MenuBars(xlWorksheet).Menus.Count
        .MenuBars(xlWorksheet).Menus(1).Delete
    Next iRemove
    For iRemove = 2 To 84
        If .CommandBars(iRemove).Visible = True Then
            .CommandBars(iRemove).Visible = False
            aRemove(iRemove) = 1
        End If
    
```

```
Next iRemove
End With
With ActiveWindow
    .DisplayGridlines = False
    .DisplayHeadings = False
    .DisplayHorizontalScrollBar = False
    .DisplayVerticalScrollBar = False
    .DisplayWorkbookTabs = False
    .DisplayZeros = False

    .Caption = ""
End With
End Sub
Public Sub Auto_Close()
    With Application
        For iRemove = 2 To 84
            If aRemove(iRemove) = 1 Then
                .CommandBars(iRemove).Visible = True
            End If
        Next iRemove
        .MenuBar(xlWorksheet).Reset
        .WindowState = xlMaximized
    End With
    With ActiveWindow
        .DisplayZeros = True
    End With
End Sub
```

Annexe B

REFSQ'2002 workshop

Dans cette annexe se trouve le document auquel j'ai collaboré à sa rédaction, notamment sur la partie concernant AHP. Ce document fera l'objet d'une étude lors de la prochaine conférence RE'02 à Essen en Septembre.

Integrating Decision-Making Techniques into Requirements Engineering

N.A.M. Maiden, P. Pavan, A. Gizikis, O. Clause*, H. Kim & X. Zhu

Centre for HCI Design
City University London

*Now with Computer Science Department, Cellule Interfacultaire de Technology Assessment, FUNDP, Université de Namur, Belgium

Abstract

Requirements engineering is an integral part of concurrent systems engineering for all types of systems from defence to business. Organisations need and use requirements to select between candidate architectures and designs throughout the concurrent systems engineering process. This emerging new role for requirements imposes new demands on requirements modelling and analysis techniques and tools. In this paper we present the results of research to extend one requirements modelling formalism, the *i** approach, with multi-criteria decision-making in the form of the AHP. We describe a software prototype called REDEPEND that supports this extension of *i**. A running example demonstrates how REDEPEND integrates *i** and the AHP supports quantitative decision-making with *i** system models from use of the AHP to inform trade-off decision-making. The paper ends with a discussion of the impact of this work on our wider ART-SCENE concurrent systems engineering environment.

1 Introduction

Requirements engineering is no longer a standalone activity in most organisations. The growth of concurrent systems engineering, in which a system's requirements, architecture and detailed design are developed at the same time, means that both the role of and challenges faced by requirements engineers have changed fundamentally. Requirements are no longer just the drivers for the design of a system's architecture. Rather, requirements provide essential tests that an evolving architecture must satisfy throughout its development. This is true for most systems no longer designed from scratch. Defence systems, from submarines to missile systems, are often built from predefined sub-systems and equipment. Likewise business systems are increasingly constructed from off-the-shelf software solutions such as SAP's R/3 and packages such as Lotus Notes. The challenges for requirements engineering research to investigate how requirements inform decision-making about the system's architecture and design. It must recognise the parallel pressures of top-down requirements push and bottom-up solution pull based on reusable solutions on requirements engineering processes, tools and techniques.

In this paper we report preliminary requirements engineering research to investigate one of the most difficult systems engineering challenges – how to use system requirements to inform decision-making to select between different candidate architectures and designs. The research is being undertaken as part of the UK EPSRC-funded ‘SIMP’ project. In SIMP we are developing a research prototype process and software tool called ART-SCENE – Analysing Requirements Trade-offs with SCENario Evaluations. ART-SCENE supports the concurrent analysis of system scenarios, requirements and architectures, and the decision-making that this analysis supports to trade-off the satisfaction of different requirements with different candidate architectures in different system scenarios. It generates outputs to a multi-criteria decision-making approach called the Inclusive Value Management, or IVM for short (M’Pherson 2001), to manage uncertainty and risk. IVM is a measuring instrument that combines measures of financial and intangible values (e.g. quality, reliability) in such a way as to indicate an overall financial cost-benefit assessment. Our use of ART-SCENE provides input data about the system’s performance to be used in performance-cost trade-off analysis.

To develop ART-SCENE we are integrating results from our own research with state-of-the-art systems engineering techniques. A team of systems engineers using ART-SCENE specifies, generates and walks through system scenarios using the CREWS-SAVRE approach reported in Sutcliffe et al. (1998) and Maiden (1998). It models systems architectures using the xUML (Kennedy-Carter 2000), a subset of the Unified Modelling Language that incorporates a complete action specification language that allows developers to build executable system models. xUML is based upon the principle of building a set of precise, testable analysis models of the system to be developed, executing defined tests on these models (Kennedy-Carter 2000). ART-SCENE incorporates part of the iUML tool suite for modelling and executing architecture models. The team also uses ART-SCENE to model system agents and requirements using *i**, a formalism for modelling complex system agents, requirements and their dependencies (Yu 1993). ART-SCENE includes REDEPEND, a software prototype for *i** system modelling that provides an effective basis for scenario generation (Boutin et al. 2002).

This paper describes research to extend the *i** requirements modelling in ART-SCENE with one multi-criteria decision-making technique, the AHP (Saaty 1990), to inform early decision-making about different architecture solutions that satisfy the modelled requirements. The remainder of the paper is in 4 sections. Section 2 describes the *i** approach and demonstrates it with a simple system selection problem. Section 3 describes the AHP technique and demonstrates it with another simple example. Section 4 describes ART-SCENE’s REDEPEND software tool in detail, and demonstrates its functions and other features with the system selection example. The paper ends with a discussion of the work so far, relates it to similar work, and outlines plans to implement and evaluate this research as part of the ART-SCENE environment.

2 Requirements Modelling with *i**

In ART-SCENE we implement the i^* approach to model complex systems and their goals. Yu (1993) developed syntax and semantics for modelling system requirements, referred to as goals in i^* . One of the key ideas in i^* is the separation of hard goals from soft goals. Hard goals equate to functional requirements that are either met or not met. In contrast, soft goals equate to non-functional requirements that different solutions can more or less contribute to. This is the crux of i^* modelling - it is possible to build up complex requirements models showing different dependencies between requirements and other concepts. These models enable the team to understand better the requirements and to be able to analyse trade-offs between the attainment of requirements by the selection of different solutions. In this section we summarise i^* 's syntax and semantics, the 2 types of i^* model, and the semantics that are important for modelling the trade-off space for a system's architecture. We are indebted to the excellent work by Eric Yu, John Mylopoulos and their colleagues at the University of Toronto.

2.1 i^* Modelling Constructs

The i^* modelling approach uses 5 basic modelling constructs. The central unit to be modelled is the intentional strategic actor. Perhaps the most important advantage of i^* modelling against other modelling approaches, such as the UML, is that an actor in i^* does not only perform actions, but has intentional aspects such as objectives, rationale and commitments. These intentional aspects associated with any actor can be distinguished into four process elements: goals, tasks, soft-goals and resources. Actors depend on each other for goals to be achieved, tasks to be performed and resources to be furnished. This section defines each of these 4 types of process element and demonstrates them with a simple example – selecting a motor vehicle to meet the requirements of an up-and-coming musician called Lawrie who needs a vehicle to provide himself and his friends with transportation to music venues.

A **goal** represents a condition or state of the world that can be achieved or not. Goals in i^* equate to functional requirements that are either met or not met. A goal of Lawrie's vehicle is that the *car has large space for transporting equipment*. A **task** represents one particular way of attaining a goal. It can be considered as a detailed description of how to accomplish a goal. Tasks are activities that produce changes in the world. In other words, tasks make conditions or states of the world to be true. One possible task of Lawrie's vehicle is to *transport goods using a trailer*.

Resources are used in i^* to model objects in the world. These objects can be physical or informational. Resources can be considered as the availability of some entity or even the finished product of some action, process or a task. The finished product of a process can be represented by a resource, which can be used by some actor. Resources that are important in the selection of Lawrie's vehicle include *fuel* (a physical resource) and the *maintenance log* (an informational resource).

The notion of goal is directly related to the notion of goal achievement. A hard goal is either achieved or not. However, there are goals that cannot be so sharply defined, such as goals that describe properties or constraints of the system being modelled. The concept

of **soft goals** in i^* can be equated to the concept of a non-functional requirement. Soft goals relevant to Lawrie's vehicle are *Low running costs* and *Reliable vehicle*.

2.2 Strategic Dependency (SD) Models

There are 2 basic types of model in i^* . The Strategic Dependency (SD) model provides a network of dependency relationships among actors. The opportunities available to these actors can be explored by matching the depender who is the actor who "wants" and the dependee who has the "ability". Since the dependee's abilities can match the depender's requests, a high-level strategic model can be developed. The main focus of the SD model is the dependencies between the actors. The Dependency Link is a link between two actors and indicates that one actor depends on another for something that is essential to the former actor for attaining a goal. The depending actor is called the depender, the actor who is depended upon the dependee and the process element around which the dependency.

There are 4 main types of dependency. For **goal dependencies**, the depender depends upon the dependee to bring about a certain state in the world. For **task dependencies** the depender depends upon the dependee to carry out an activity. For **resource dependencies**, the depender depends upon a dependee for the availability of an entity. For **soft goal dependencies**, the depender depends upon the dependee to perform some task that meets the soft goal or to perform the task in a particular way.

2.3 Strategic Rationale (SR) Models

The second type of i^* model is the Strategic Rationale (SR) model. The SR model provides an intentional description of processes in terms of process elements and the relationships or rationales linking them. A process element such as a task or resource is included in the SR model only if it is considered important enough to affect the achievement of some goal. The SR model includes the SD model and hence actors may be able to accomplish something by themselves, or by depending on other actors.

We can link the four types of nodes (**goals, tasks, resources and soft goals**) using any of the 4 available relationship links available in the SR model. These are the **Dependency Link** (the same as in the SD model), the **Task Decomposition Link**, the **Means End Link**, and the **Contribute To Soft Goal Link**. We describe the Task Decomposition and Means End Links in this section, then explore an innovative role of Contribute To Soft Goal Links in the next section.

A **task** is modelled in terms of its **decomposition** into sub-components, represented by the already defined process elements. A task specifies how to achieve a desired state, so when there is enough and decomposable knowledge of how to perform a task, then this knowledge can be decomposed into sub-components (sub-processes) by the task decomposition link. Since all the sub-components need to be completed for the task to be performed, the relationship between the sub-components can be interpreted as a logical

“AND”. For example, the earlier task *transport goods using a trailer* can be decomposed into 3 sub-tasks, *attach trailer*, *pull trailer* and *detach trailer*. Task decomposition links can also be seen in Figure 1 – the tasks transport, contain passengers and contain luggage are all decomposed into goals, soft goals and resources using task decomposition links. It is these decomposition links, and the resulting hierarchical structures, that make the application of multi-criteria decision-making techniques such as the AHP feasible.

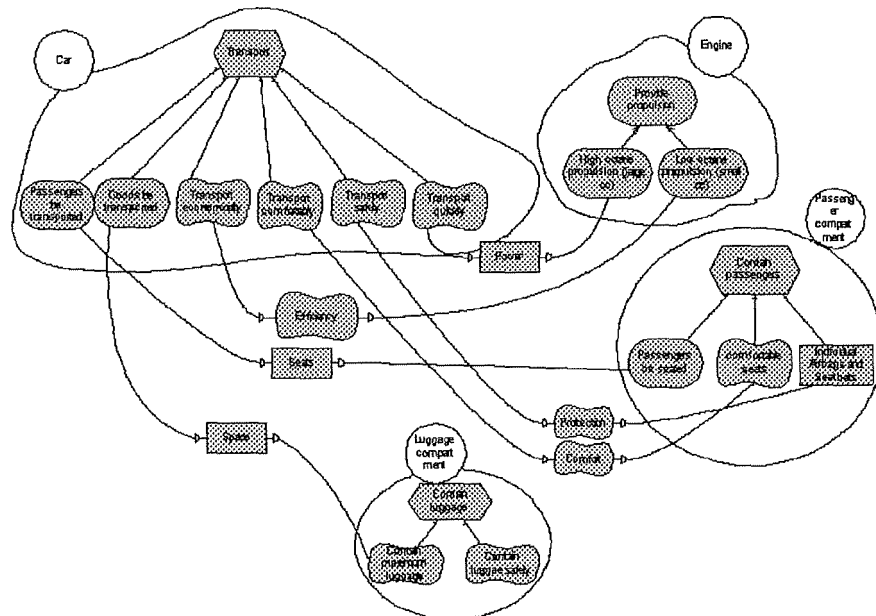


Figure 1. A Strategic Rationale model for Lawrie's vehicle, showing the hierarchical decomposition of goals and tasks for the different system agents

The **Means End Link** indicates a relationship between an end and a means for attaining the end. Each Means End Link provides a different mean for attaining the end. Whenever there are different ways to achieve a goal, the Means End Link should be used to represent this relationship, which acts as a logical ‘OR’ between the different ways (means) of accomplishing the goal. The Means End Link is graphically represented with an arrow, where the arrowhead points from the means to the end. In our example of Lawrie's vehicle, the engine can provide propulsion either by *providing high octane propulsion* or by *providing low octane propulsion*, as also shown in Figure 1.

2.4 Contribute-To Soft Goal Links

A **Contribute To Soft Goal Link** is a Means End Link with a soft goal as the end. Since the achievement of a soft goal cannot be clearly defined, any means end link with a soft goal as the end, would be a confusing concept. This is due to the fact, that no means can be specified for the achievement of a goal (end) that cannot be clearly defined. However, a slightly different kind of relationship link can be defined and this is the Contribute To Soft Goal Link. This link represents that a goal, a task, a resource or a soft goal can be positively or negatively contributing to the attainment of a soft goal, without

ensuring the attainment. For example, the use of a *low octane (small cc engine)* can contribute positively to achieving the soft goal *cheap insurance costs*, and it contributes negatively to achieving the soft goal *arrive quickly*.

Furthermore, in ART-SCENE, Contribute To Soft Goal links, and in particular negative contribution links between soft goals, have an important and innovative role in scenario generation. Negative contribution links define trade-offs between non-functional requirements that we need to optimise for the whole system when selecting the final system architecture. They are, in simple terms, a specification of the trade-off space in terms of system requirements that is independent of the architecture. In ART-SCENE we use these negative contribution links to guide the generation of scenarios designed to exercise candidate architectures in the design trade-off space. Users of ART-SCENE are able to focus their analytic processes on the trade-offs that really matter for the system. Therefore we envisage a bi-directional relationship between MCDM techniques such as the AHP and scenario-driven trade-off analyses in ART-SCENE. The AHP provides systems engineers with a simple-to-use decision-making technique to inform high-level trade-offs. These trade-offs are then explored further through scenario analyses, and the results of these analyses provides further data with which to undertake more rigorous MCDM analyses.

In the next section we introduce and describe the AHP multi-criteria decision-making technique that can support systems engineers to make trade-offs in such a design space.

3 Multi-Criteria Decision-Making with the AHP

Multiple criteria decision-making (MCDM) refers to making decisions in the presence of multiple, usually conflicting criteria. Although MCDM problems are widespread, MCDM as a discipline has a relatively short history of about 30 years (Xu & Yang 2001). Its development is closely related to the advancement of computers. The rapid development of computer technology in recent years has made it possible for the first time to conduct systematic analyses of complex MCDM problems. Furthermore, the widespread use of computers and information technologies has generated more information that can be used in decision-making with MCDM techniques.

There are two basic types of MCDM methods – **compensatory** and **non-compensatory** (Hwang & Yoon 1981). **Non-compensatory methods** do not permit trade-offs between the attributes of different candidates to choose between. An unfavourable value in one attribute cannot be offset by a favourable value in other attributes. The assumption is that each attribute (for example a requirement or a solution characteristic) is independent of all other attributes, so that comparisons are made on an attribute-by-attribute basis. The MCDM methods in this category are credited for their simplicity, however most complex systems do not have independent requirements – agents depend on each other, the satisfaction of one requirement contributes to the satisfaction of another requirement, and architecture components are inter-connected, as *i** models highlight. Such methods are too simplistic for decision-making about requirements-architecture trade-offs.

In contrast, **compensatory methods** permit decision-makers to make trade-offs between attributes. A decline in one attribute is acceptable if it is compensated by some enhancement in one or more other attributes. Such compensations are commonplace in software and systems engineering – engineers make often implicit decisions to trade-off the performance, cost, reliability and time-to-deliver of a new system. Several compensatory MCDM methods have been published, however in ART-SCENE we have chosen the Analytic Hierarchy Process, or AHP for short (Saaty 1988), to inform trade-off decision-making. There are 2 reasons for this choice. Firstly, the AHP arranges all decision factors (i.e. system goals) in a hierarchical structure, which descends from an overall goal to criteria, sub-criteria and finally to the alternatives, in successive levels. This structure maps on to traditional requirements structures including some structures that are supported in i^* SR models. Secondly, the AHP's pair-wise comparison of decisions factors and alternatives in matrices overcomes one important limitation of other MCDM methods, that we can compare alternatives against decision factors, or requirements, of different types. ART-SCENE implements a taxonomy of over 20 different types of requirement (from functional to performance and training type), so the indirect pair-wise comparison becomes an essential feature of any measurement approach.

The AHP method is not new to software engineering research. Kontio (1996) applied AHP for software package selection, and more recently the SCARLET process integrates the AHP with requirements evaluation techniques to enable software component selection as part of the EU-funded BANKSEC project (Maiden & Kim 2002).

In the remainder of this section we describe a simple example to demonstrate the AHP method. The example defines 4 candidates C1, C2, C3, and C4 to chose from, and 3 decision factors (or requirements) R1, R2 and R3 that will be used to choose between the candidates. We assume that R1, R2 and R3 are 3 leaf-nodes requirements of a simple hierarchy that decomposes the higher-level requirement R. A decision-maker applies the AHP method to this problem in 3 stages:

- (1) Pair-wise comparison of candidates against each decision-factor, to rank the candidates against the factor;
- (2) Pair-wise comparison of the decision factors themselves, to provide a relative ranking of all of the decision factors in the hierarchy;
- (3) Computation of candidate rankings for all decision factors using the candidate rankings from (1) and the criteria rankings from (2).

3.1 Pair-wise Comparison of Candidates

In stage (1) the decision-maker compares each pair of candidates from C1, C2, C3 and C4 against each decision factor R1, R2 and R3 in turn. One matrix is produced for each factor, so there will be 3 matrices, one for R1, one for R2 and one for R3. By convention, the comparison of strength is always of an alternative in the left-hand column against an

alternative listed across the top. Each number in the matrix expresses the result of a pairwise comparison between 2 candidates. Each number is a quantitative approximation that expresses the comparative strength of the comparison where, for 2 candidates C1 and C2 for decision factor R:

- C1 and C2 are equally important for achieving R, **insert 1**
- C1 is weakly more important than C2 for achieving R, **insert 3**
- C1 is strongly more important than C2 for achieving R, **insert 5**
- C1 is demonstrably or very strongly more important than C2 for achieving R, **insert 7**
- C1 is absolutely more important than C2 for achieving R, **insert 9**

in the position (C1,C2) where the row of C1 meets the column of C2. A candidate is equally important when compared with itself, so where the row of C1 and column of C1 meet in position (C1,C1) insert 1. Thus the main diagonal of a matrix must consist of 1s. We insert the appropriate reciprocal 1, 1/3, 1/5, 1/7 or 1/9 where the column of C1 meets the row of C2, i.e., position (C2, C1) for the reverse comparison of C2 with C1. The numbers 2, 4, 6, 8 and reciprocals are used to facilitate compromise between slightly different judgements. Table 1 shows possible judgements for the decision factor R1.

	C1	C2	C3	C4
C1	1	3	1/5	1/7
C2	1/3	1	1/6	1/9
C3	5	6	1	1
C4	7	9	1	1
Sum	13.33	19.00	2.37	2.25

Table 1. A possible judgement matrix for the decision factor R1. C1 is weakly more important than C2 for satisfying decision factor R1, so it receives a score of 3, while C4 is absolutely more important than C1 for satisfying decision factor R1, so it receives a score of 9

Once the matrix for each decision factor R has been completed the decision-maker normalizes the judgement values by dividing each value by the sum of the values, as shown in Table 2. The decision-maker inserts the resulting priorities, that is the relative ratings of each candidate, in a new right-hand column of each matrix. These priorities are the averages of the values in that row, for example candidate C1's priority value is computed by reading left-to-right along the C1 row: $(0.08+0.16+0.08+0.06)/4 = 0.095$.

	C1	C2	C3	C4	Priority
C1	0.08	0.16	0.08	0.06	0.095
C2	0.03	0.05	0.07	0.05	0.049
C3	0.38	0.32	0.42	0.44	0.389
C4	0.53	0.47	0.42	0.44	0.466
Sum	1.00	1.00	1.00	1.00	1.00

Table 2. The equivalent normalised matrix for the decision factor R1. The priorities suggest that the preferred candidate C4, followed by C3, then C1 and C2.

In a full presentation of the example, the decision-maker would complete similar judgement matrices for decision factors R2 and R3, then normalise them in the same manner.

3.2 Pair-wise Ranking of Criteria

In stage (2) the AHP provides a similar method for ranking the decision factors to make more informed decisions about each candidate's overall ranking for the highest-level decision factor R. A matrix is developed for each level and branch of the hierarchical decomposition of decision factors. Again a pair-wise comparison of criteria is undertaken using a similar mechanism for allocating judgement values to each decision factor pair. The qualitative judgements and corresponding scores differ slightly due to the different nature of each comparison. Each number is a quantitative approximation that expresses the comparative strength of the comparison where if, for 2 decision factors R1 and R2:

- R1 and R2 are equally important, **insert 1**
- R1 is weakly more important than R2, **insert 3**
- R1 is strongly more important than R2, **insert 5**
- R1 is demonstrably or very strongly more important than R2, **insert 7**
- R1 is absolutely more important than R2, **insert 9**

Table 3 demonstrates a simple criteria ranking matrix for the decision factors in the example, and we show the equivalent normalised matrix, with the calculated priority value for each criterion shown in the right-hand column of Table 4.

	R1	R2	R3
R1	1	1/5	1/2
R2	5	1	5
R3	2	1/5	1
Sum	8.00	1.40	6.50

Table 3. A possible judgement matrix of the 3 decision factors R1, R2 and R3. For example decision factor R2 is strongly more important than R1, so it receives a score of 5.

	R1	R2	R3	Priority
R1	0.13	0.14	0.08	0.115
R2	0.63	0.71	0.77	0.703
R3	0.25	0.14	0.15	0.182
Sum	1.00	1.00	1.00	1.000

Table 4. The equivalent normalised matrix for the 3 decision factors R1, R2 and R3. The resulting priorities scores reveals that R2 is the most important decision factor, followed by R3 then R1.

3.3 Calculating Candidate Scores

The third stage (3) combines the results from the previous 2 stages to produce an overall ranking for each candidate C against all of the decision factors. Again a simple matrix

listing all decision factors R and candidates C is produced. Furthermore an additional column showing the calculated priorities of each criterion R is included, as shown in Table 5. The decision-maker puts the calculated priority of each criterion in the 2nd column, then inserts the priorities accordingly. For example, the priorities of candidates C1, C2, C3 and C4 with respect to the decision factor R1 were 0.095, 0.049, 0.389, and 0.466. To calculate the overall score of each candidate, the decision-maker multiplies each priority and its corresponding weight, then sums them all. The candidate C1 has the priorities 0.095, 0.639, and 0.052 for the criteria A1, A2, and A3. Therefore C1's overall priority is $0.115 * 0.095 + 0.703 * 0.639 + 0.182 * 0.052 = 0.469$. This application of AHP method advises the decision-maker to select the component C1 as it achieved the highest overall score. A ranking of the alternatives is achieved by a procedure that vertically calculates the horizontal comparison ratios. The AHP assumes that it is actually possible to determine the difference between the decision criteria's importance and the alternatives' performances in the form of meaningful numeric ratios (Saaty 1988).

	Priority	C1	C2	C3	C4
R1	0.115	0.095	0.049	0.389	0.466
R2	0.703	0.639	0.056	0.132	0.173
R3	0.182	0.052	0.548	0.256	0.145
Weighted Sum		0.469	0.145	0.184	0.202

Table 5. The Final Weighted Sum Matrix for the Simple Example.

Obviously the completion of some many matrices can lead to potentially-inconsistent pair judgements by the decision-maker. The AHP method handles inconsistencies regarding transitivity within matrices by applying its *eigenvalue* method. There are other possible drawbacks of the AHP. First of all, because of the pair-wise comparison approach, the AHP does not produce independent ratings – other techniques implemented in ART-SCENE provide direct measures of requirement-architecture compliance. Secondly, the inclusion of new alternatives and criteria can require the repetition of pair-wise comparisons for re-establishing a ranking order – a time-consuming and laborious process. Finally, the quantification of qualitative, subjective pair-wise comparisons can sometimes suggest a level of precision that was not actually available from the AHP method (Girod 2000). Therefore, the AHP method needs to be applied and interpreted with some care, but we believe that it provides a simple and useful approximation tool in the early stages of systems engineering.

4 The REDEPEND Prototype

Our research integrates *i** requirements modelling and AHP multi-criteria decision-making in our ART-SCENE environment, and in particular in one of ART-SCENE's modules called REDEPEND. REDEPEND is a graphical modelling tool developed as a plug-in to Microsoft Visio 2000 that enables a user to construct and analyse *i** SD and SR models. Version 1 of REDEPEND has been used extensively to produce *i** models of a European air traffic management system as part of the CORA-2 project (Maiden et al. 2002). In ART-SCENE we use the *i** SD and SR models for 3 specific purposes:

1. More effective modelling of requirements for complex systems that are composed of interacting agents;
2. A reuse-driven approach to scenario authoring and generation with ART-SCENE's Scenario Palette tool (Boutin et al. 2002);
3. First-cut system models for exploring high-level requirements-architecture trade-offs using scenario-based techniques.

Our integration of the AHP into REDEPEND supports the third purpose – to explore possible requirement-architecture trade-offs before the use of scenarios, then to guide the generation of scenarios that explore these trade-offs further.

4.1 Integrating i^* and the AHP in ART-SCENE

Although useful for the early modelling of system requirements, the informal nature of i^* system models can sometimes make them difficult to use for more rigorous and quantitative decision-making that is essential during concurrent systems engineering. Indeed, our use of i^* in systems engineering organisations has been criticised for the lack of numbers indicating the degree of goal-task compliance and goal rankings. Therefore we argue that integrating i^* models with the AHP also introduces greater precision in model analysis and model data upon which to make requirement-architecture trade-off decisions. To achieve this we have designed REDEPEND not only to support trade-off decision-making but also to annotate existing SD and SR models with quantitative data resulting from the application of the AHP method.

We have integrated i^* and the AHP using the mappings between concepts in i^* and the AHP shown in Table 6. Some mappings are intuitive and obvious, others less so. A decision factor in the AHP maps to an i^* goal and soft goal – goals and soft goals equivalent to system requirements are the explicit selection criteria in complex decision-making. The hierarchical structure of criteria essential to the application of the AHP is represented in i^* using means-end links between goals that are structured to form goal decomposition hierarchies. The lack of such hierarchies is an important constraint on the application of the AHP – a constraint that we apply automatically in REDEPEND. A candidate or alternative in the AHP maps to a task in i^* . This is not intuitively appealing (many systems engineering solutions are not tasks), but tasks are the only representation available for modelling the solution space in i^* . A positive contribute to goal link to a soft goal maps to a high priority rating for that soft goal – the means defined in the link contributes positively to the satisfaction of the goal. Likewise, a negative contribute to goal link to a soft goal maps to a low priority rating for that soft goal – the means defined in the link contributes negatively to the satisfaction of the goal.

i^* Modelling Element	AHP Element
Goal and soft goal	Decision factor
Means-end links in hierarchical structure	Hierarchies of decision factors
Task	Candidate
Positive contribute to soft goal link	High priority rating

Negative contribute to soft goal link	Low priority rating
---------------------------------------	---------------------

Table 6: Mapped concepts in i^* and the AHP.

4.2 REDEPEND's Basic Features

Figure 2 shows an example of REDEPEND being used to develop a SR model for a submarine system. The model shows 4 agent types – submarine, manoeuvring control system, actuation systems and hydroplanes – and the key goal and task structures and dependencies for these 4 agents. Different i^* modelling constructs are shown on the drawing palette on the left-hand side. The user is able to drag and drop the required elements into the work area and use the pre-defined links to provide the associations between the different elements within a model. The right-hand mouse button gives access to i^* -specific functions such as changing the type of an object (e.g. from goal to soft goal), adding additional information to each model object, and checking model properties. The i^* modelling approach includes rules and restrictions on the associations between elements in SD and SR models, and we have implemented these rules and restrictions in REDEPEND.

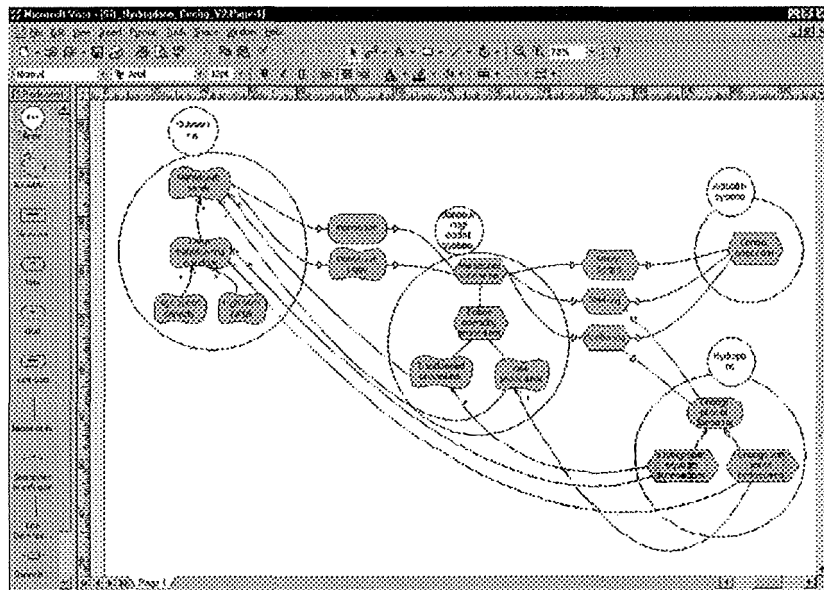


Figure 2. An Example of REDEPEND being used to produce a SR model.

REDEPEND also provides user-centred functions that support both beginner and advanced users of i^* . Beginners have the option of enabling REDEPEND to verify new connections that are drawn onto a model against i^* rules and regulations, thus reducing the modelling learning curve and providing reference material for manual model checking. Later, when a user becomes more experienced with i^* modelling, automatic verification of connections can be switched off. In the event of a rule-violation, the user is presented with a three-part error message such as that shown in Figure 3 describing the error and a suggestion about an alternative connection type. In all cases the user can override the tool's error warnings and suggestions. REDEPEND also has functions to log system errors and import them into an error log database.

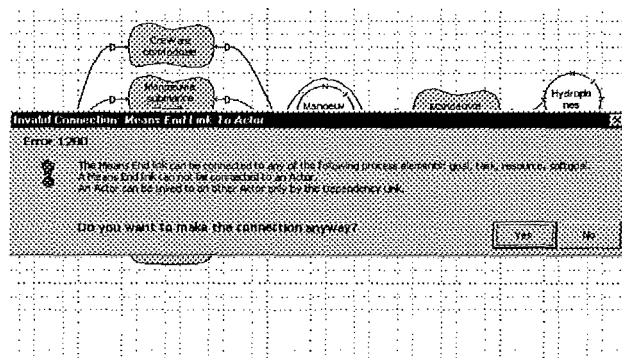


Figure 3. Example of model checking advice from REDEPEND.

Users can also customise REDEPEND to individual projects by inserting predefined *i** elements in templates applicable to all models for the project. In the example shown in Figure 3, a template can include predefined actors such as a hydroplane or an actuation system for reuse in different SD and SR models. Furthermore, REDEPEND enables the user to store agent goal, soft goal, task and resource descriptions to elaborate the models and provide reusable data within the models. Figure 4 shows a task description comprising 6 actions for the *detect boat* task of the *i** model.

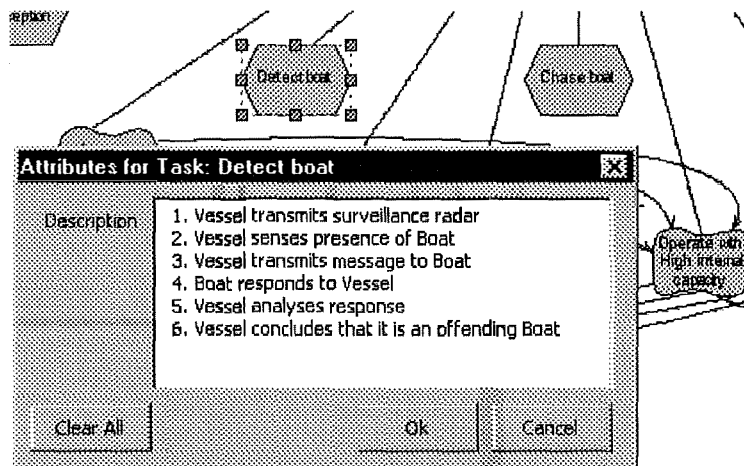


Figure 4. Example task description templates.

4.3 REDEPEND's AHP Component

We have added a new AHP feature to REDEPEND to support the AHP method with *i** modelling. We intend that users will use the AHP feature to explore high-level decisions early in the systems engineering process to inform selection and trade-off decisions about coarse-grain alternatives. We have implemented the feature by extended REDEPEND with Microsoft Excel spreadsheets that implement the AHP method. To apply the AHP method a user selects a goal or a soft in the goal hierarchy to open a pair-wise comparison matrix. If the selected goal or soft goal is decomposed into further goals and soft goals, then the matrix supports the ranking of these goals or soft goals. If the selected

goal or soft goal is a end of a means represented by one or more tasks, then the matrix supports the ranking of different task solutions to meet that goal.

REDEPEND automatically imports the goals (decision factors) and tasks (alternatives) into the matrix, which opens automatically for the user to complete, as shown in Figure 5. The user can enter comparison scores directly into the matrix or be guided by the pair-wise comparison wizard. The matrix calculates the priority matrices for each decision factor or alternative and, at the end of the pair-wise comparisons, stores all pair-wise comparison scores, and imports the resulting priorities into the i^* model by labelling the goal and soft goal decomposition links and the means-end links. Therefore further changes to any of the AHP matrices can be propagated automatically to the i^* model.

Figure 5. REDEPEND's AHP pair-wise comparison matrix and wizard.

In the remainder of this section we will demonstrate the integrated i^* modelling and AHP method to help Lawrie chose a vehicle that best meets his needs and the needs of his friends.

4.4 Using REDEPEND to Select Best-Fit Alternatives: An Example

Let us return to Lawrie and his vehicle selection problem – after all – there is a new gig coming up. Lawrie is an up-and-coming musician and has always relied on someone else to transport his equipment and friends to the music venues. Recently, he has been considering by a vehicle of his own to provide transportation for his equipment and friends. However, the available choices are overwhelming him, some of his decision

factors conflict (for example reliable cars tend to cost more money), and he will probably have to make some complex trade-off decisions. He does not know whether to choose a new van, a used van, a new car, a used car or to rent a vehicle as needed. Therefore he decides to use REDEPEND to model his requirements and choices.

Lawrie starts to develop a SR model. His most important goal is to *Be transported*. He knows of different candidate means of achieving this goal, including transporting with a new van, with a used van, with a used car, with rented vehicles and with new a car. The first-cut SR model is shown in Figure 6. Lawrie has a clear preference, to be transported using a new car. This straightforward decision-making does not need the AHP method, so he continues to develop the SR model by decomposing the *transport with new car* task.

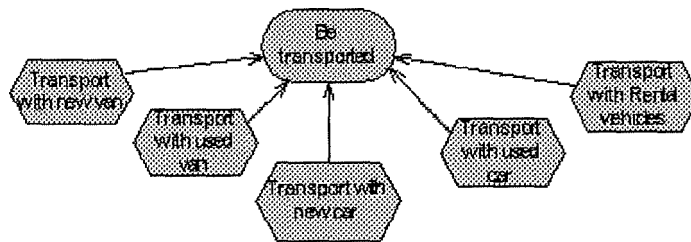


Figure 6. The initial SR model, showing different means to achieve the goal *Be Transported*.

Lawrie identifies 6 goals and soft goals when being transported using a new car. The car shall have 5-seat capacity, low running costs, adequate luggage capacity, and be safe, reliable, comfortable and affordable. He extends the SR model as shown in Figure 7 using task decomposition links. He also explores the new car market, modelled as tasks in i^* . He could buy a car with large passenger and luggage space, a cheap car with a proven safety record but higher running costs, or a small car with a small engine with little luggage space. Candidate vehicles include a *Volvo 850*, *Lada Niva*, *VW Golf* and *BMW SUV*. Each vehicle contributes positively and negatively to one or more of Lawrie's soft goals, as also shown in Figure 8. For example, *buying a BMW SUV* contributes positively to the satisfaction of the soft goal *car must be comfortable*, but it contributes negatively to the soft goal *car must be affordable*.

Figure 7. The extended SR model showing goals and soft goals that decompose the high-level task, and candidate solution vehicles that contribute positively or negatively to each of these goals and soft goals.

Although i^* 's contribute-to links provide important evidence about the overall degree of satisfaction of the goals and soft goals by each alternative, the attribution of positive and negative contribution links is subjective. Furthermore the SR model does not prioritise the goals and sub-goals or attribute costs to the different solutions. He needs more information to make his selection. Therefore he applies the AHP method to this multi-criteria decision-making problem.

4.5 Pair-wise Comparison of Goals and Soft Goals

First of all, Lawrie requests REDEPEND to create an AHP matrix for the pair-wise comparison of the 6 requirements in order to rank them. He selects the *transport with new car* task and initiates the AHP feature. REDEPEND opens a blank 6x6 matrix with the goal and soft goal names automatically imported into the matrix rows and columns, as shown in Figure 8. The AHP wizard prompts the user to make the pair-wise comparisons by requesting answers to questions such as:

- 1 Adequate luggage capacity in relation to safe and reliable car?
- 2 Car must be comfortable in relation to car must be affordable?

in turn, until the matrix is filled. REDEPEND calculates the normalised priority values and displays them in the right-hand column of the matrix.

Figure 8. REDEPEND's AHP pair-wise comparison feature, with: (a) the direct form-fill matrix; (b) the pair-wise comparison matrix, and; (c) the total priority calculations for each goal and soft goal in the matrix.

Lawrie chooses to import the results from the AHP analysis into the SR model and display them as attributes of the relevant task decomposition links, as shown in Figure 9. For example, the matrix shown in Figure 8 shows that the soft goal *Safe and reliable car* has the highest priority ranking (0.32), the goal *5-seater capacity* has the second-highest ranking with a score of 0.26, and the soft goal *Adequate luggage capacity* has the third-highest ranking with a score of 0.23. The scores are shown on the related SR model in Figure 9.

Figure 9. Part of the SR model showing the results of the AHP method to prioritise goals and sub-goals that decompose the high-level task.

4.6 Pair-wise Comparison of Alternatives

Next Lawrie uses the AHP feature to determine how well each candidate alternative, i.e. each vehicle satisfies the goals and soft goals. He uses REDEPEND in a similar way. He completes one matrix for each requirement by undertaking pair-wise comparisons of the 4 alternatives for each goal and sub-goal, as shown in the matrix in Figure 10. For example, the matrix shows the pair-wise comparison matrix for one of the soft goal *safe and reliable* car. Lawrie decides that the BMW SUV is a little more important than the Volvo 850 in satisfying the *safe and reliable* soft goal so it is given a score of 2. In contrast the BMW SUV is absolutely more important at satisfying this soft goal than the Lada Niva, and is given a score of 9.

Figure 10. REDEPEND's AHP pair-wise comparison feature showing the direct form-fill matrix for one of Lawrie's sub-goals.

The resulting SR model, with ranking attributes specified for each task to goal or sub goal contribution link on the model, is shown in Figure 11. Lawrie's final step is to use REDEPEND to calculate the overall score that indicates how well each alternative enables him to undertake the highest-level task, *transport with new car*. The resulting rankings indicate the overall level of satisfaction of the high-level task by each candidate solution. He does this by selecting the task *transport with new car* and selecting the AHP function from the context menu. REDEPEND performs the AHP calculation and displays the resulting AHP priority values adjacent to the candidate solutions, as shown in Figure 11.

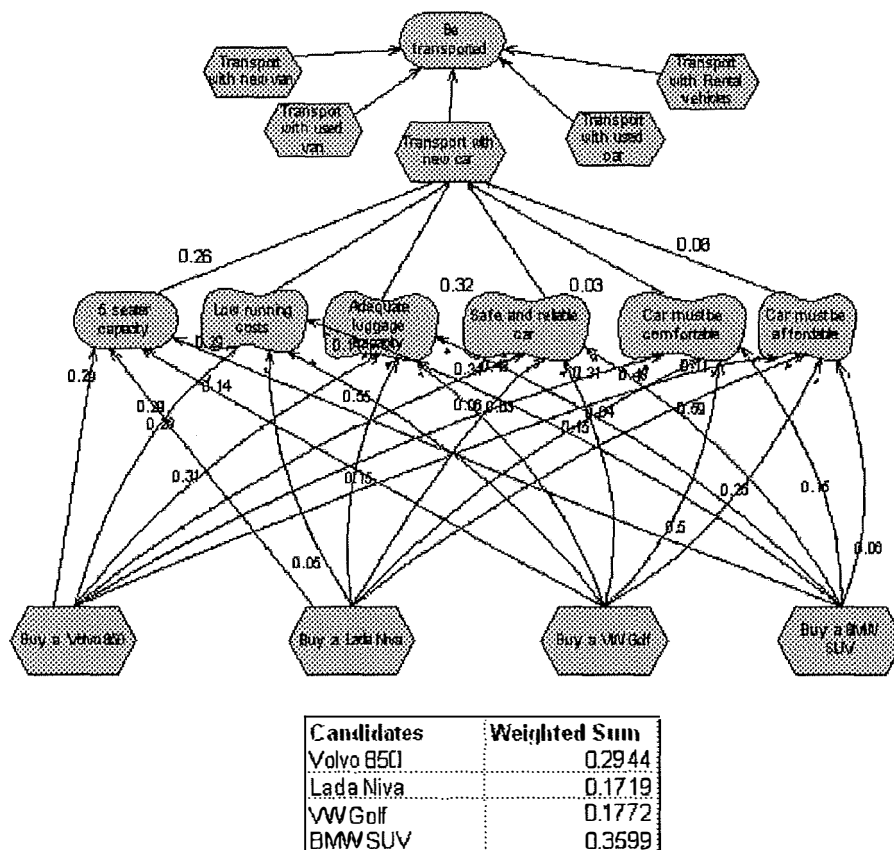


Figure 11. The final SR model, showing the final AHP priority values for each of the 4 solutions with respect to the high-level task *transport with new car*.

Figure 11, Lawrie's final SR model, reveals the results of the AHP method for this decision-making exercise. The method calculates that the highest rating priority score is for the *BMW SUV* then the *Volvo 850*, hence REDEPEND recommends the *BMW SUV* to Lawrie. Lawrie can then use this SR model and the completed matrices to undertake sensitivity analyses, as long as the SR model itself does not change. For example he can change the ratings of the goal and sub-goal priorities and the pair-wise comparisons of each candidate alternative with one or more of these goals and sub-goals, and REDEPEND will propagate these changes through the SR model to change the final ranking scores for each candidate alternative. This simple environment for exploring requirement-solution trade-offs represents an important and novel advance for requirements modelling environments.

5 Future Work in ART-SCENE

The current version of REDEPEND implements the AHP feature demonstrated with the Lawrie example. So far we have tested the AHP feature with several multi-criteria decision-making problems – as such it represents research-in-progress. A more comprehensive re-design of REDEPEND, version 2.0, will be released in June 2002 and

will include further enhancements to the AHP feature. We hope to be able to report these enhancements in the near future. We will also report the further integration of REDEPEND into the ART-SCENE environment.

The next stage of the research is to have third-party systems engineers use REDEPEND version 2.0 in system modelling and decision-making trials. Two such trials are planned. In the first we will deliver REDEPEND to systems engineers at Eurocontrol to use it to produce and revise a suite of system models for a future air traffic control system that supports conflict resolution by air traffic controllers. These engineers have worked with version 1.0 and are well placed to provide both quantitative and qualitative feedback on new REDEPEND features. We also anticipate that version 2.0 will also be used in several new Eurocontrol projects developing new air traffic control systems. The second trial will be with BAE SYSTEMS as part of the EPSRC-funded SIMP project. Two groups of BAE SYSTEMS engineers will use REDEPEND for different tasks. The first group will use the AHP feature to extend an existing pattern language about submarine manoeuvring systems represented using i^* in REDEPEND with quantitative data derived using the AHP method. The second group of systems engineers will use REDEPEND to model a complex new naval platform using i^* . We will encourage them to explore high-level decisions about this platform design using the AHP feature. For both groups we will undertake evaluations to capture data about the utility and usability of REDEPEND in general and the AHP feature in particular.

Current work has already reveal 3 important extensions to both i^* and REDEPEND. Firstly the i^* approach's current reliance on tasks as the only representation of a solution (Yu 1993) is inadequate for complex systems of systems. Therefore we will provide a local extension to i^* implemented in REDEPEND to model systems engineering solutions such as an architecture or equipment sub-system. Secondly we will extend REDEPEND's AHP feature to provide more exploration functions to undertake sensitivity analyses with SR models using the AHP approach. Thirdly, we will integrate our combined requirements modelling and multi-criteria decision-making method in REDEPEND with the IVM (M'Pherson 2001), another multi-criteria decision-making approach that manages uncertainty and risk in systems engineering projects. We use the IVM as a wider measuring instrument that combines measures of financial and intangible values to indicate an overall financial cost-benefit assessment for a system. Our REDEPEND approach outputs values of system performance (with respect to the satisfaction of system requirements) that provide input values into the IVM. We are currently developing an integrated SIMP process to support decision-making at different stages in the systems engineering process, and integrated tool support to pass system performance values automatically between tools, thus providing a more integrated and interactive systems engineering environment.

Acknowledgements

The authors wish to thank all at Eurocontrol and BAE SYSTEMS for their feedback on the REDEPEND environment. Part of this research is funded by the EPSRC Systems Integration Initiative SIMP project.

6 References

- Boutin M., Pavan P. & Maiden N.A.M., 2002, 'Composing Scenarios: Painting a Picture with the Scenario Palette', submitted as short paper to RESFQ'2002.
- Girod M., Elliott A.C. & Wright I. C., 2000, 'Decision-making and Design Concept Selection', Proceedings of the Engineering Design Conference, 659-666, Brunel University, June 2000.
- Hwang C. L. & Yoon K., 1981, 'Multiple Attribute Decision Making Methods and Applications', Springer-Verlag.
- Kennedy Carter, xUML – Executable Modelling with the UML, 2000.
- Kontio J., 1996, 'A Case Study in Applying a Systematic Method for COTS Selection', Proceedings 18th International Conference on Software Engineering, IEEE Computer Society Press.
- M'Pherson P. K. M., 2001, 'RACE for Faster, Better, Value', Proceedings International Systems Engineering Conference INCOSE'2001.
- Maiden N.A.M., 1998, 'SAVRE: Scenarios for Acquiring and Validating Requirements', *Journal of Automated Software Engineering* 5, 419-446.
- Maiden N.A.M. & Kim H., 2002, 'SCARLET: A Requirements Engineering Process for Selecting Components', submitted to journal.
- Maiden N.A.M., Robertson S. & Gizikis A., 'Cooking Up Requirements: How to Provoke Creative Requirements Engineering', Technical Report, Centre for HCI Design, City University London, April 2002.
- Saaty T.L., 1988, 'The Analytic Hierarchy Process, University of Pittsburgh.
- Sutcliffe A.G., Maiden N.A.M., Minocha S. & Manuel D., 1998, 'Supporting Scenario-Based Requirements Engineering', *IEEE Transactions on Software Engineering*, 24(12), 1072-1088.
- Xu L. & Yang J.B., 2000, 'Introduction to Multi-Criteria Decision Making and the Evidential Reasoning Approach', Working Paper Series 2001, Document Number 0106, Manchester School of Management, UMIST, UK.
- Yu E.S.Y, 1993, 'Modelling Organisations for Information Systems Requirements Engineering', Proceedings 1st IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, 34-41.

Addenda au chapitre 5 et chapitre 6

1 Addenda au chapitre 5

1.1 Comment peut-on intégrer AHP dans le modèle i* ?

Il est apparu évident que le modèle AHP pouvait facilement s'adapter au modèle i*
L'adaptation que j'ai faite de l'outil redepense s'explique comme ceci¹ :

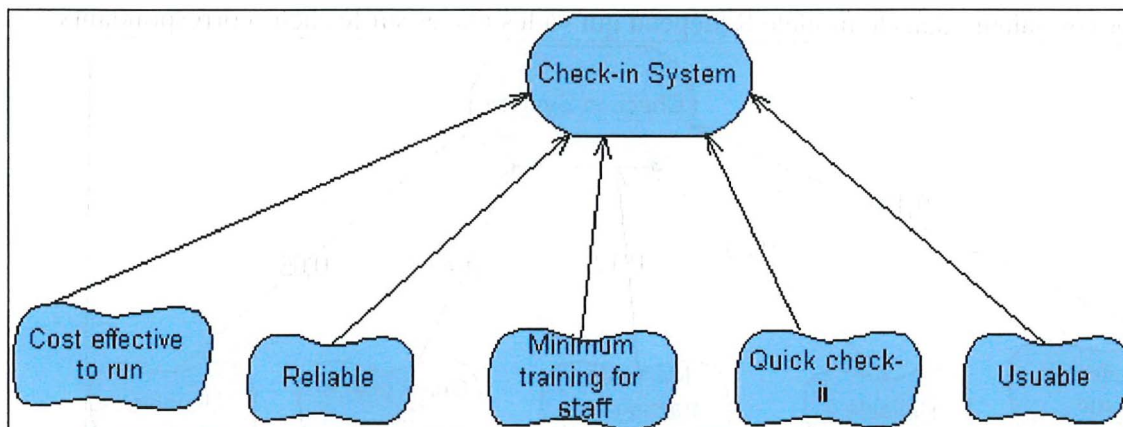
La théorie AHP suggère 3 niveaux.

- Le niveau 1 désignant l'objectif général
- Le niveau 2 désignant les critères de décision
- Le niveau 3 désignant les alternatives

Nous pouvons donc utiliser les éléments du modèle i* à savoir les goals, softgoals ou tasks pour représenter ces différents niveaux. Le calcul AHP se déroule en 3 étapes. Le choix de 3 étapes résulte du fait que cela concorde avec la méthode AHP et que cela permet aussi de montrer les étapes intermédiaires ou d'utiliser AHP dans l'optique niveau de contribution des sous-objectifs à un objectif.

1^{ère} étape

Dans la première étape il faut représenter l'objectif principal ainsi que tous les critères de jugements qu'on a définis préalablement. Ces critères concordent bien avec la définition de soft goal, c'est pourquoi on va donc utiliser cette notion pour les représenter dans le modèle i*.



Une fois cela fait, il faut maintenant se demander comment construire les matrices de critères et alternatives pour chaque élément.

Pour la matrice de critère il suffit de relier les critères de décision avec un lien « Means-end ».

¹ Pour de plus amples informations à propos de l'outil et une description mathématique détaillée veuillez vous référer au document en annexe » The AHP method and the i* Model »

ubs 10076851

Redepend fait alors appel à une routine que j'ai développée avec Microsoft Excel pour le calcul matriciel.

Pour se faire, il suffit d'appeler la fonction Excel à partir de l'objectif principal.

On obtient alors une matrice carrée reprenant en colonne et en ligne les noms des critères de décision. Une fenêtre interactive apparaît alors reprenant les valeurs de l'échelle de Saaty afin de compléter chaque cellule de la matrice avec la bonne valeur.

The screenshot shows the 'Ahp' software window. On the left is a sidebar with 'FACULTES UNIVERSITAIRES N.-D. DE LA PAIX NAMUR' and a 'Bibliothèque' button. The main area displays a comparison matrix for five criteria: Cost effective, Usuable, Minimum training, Reliable, and Quick check-in. To the right, a 'result' table shows the calculated weights (WEIGHTS) for each criterion.

	Cost effective	Usuable	Minimum training	Reliable	Quick check-in
Cost effective	1	5	3	1/3	1/3
Usuable	1/5	1	1/5	1/5	1/5
Minimum training	1/3	5	1	1/3	1/3
Reliable	3	5	3	1	1
Quick check-in	3	5	3	1	1

SR	WEIGHTS
0,89578424	0,18
0,23364746	0,33
0,61072839	0,12
1,62991996	0,33
1,62991996	0,33

Buttons: Wizard, Import into Visio

The 'matrix...' dialog box allows selecting a value for the comparison 'Cost effective to run in relation to Usuable'. It features a horizontal scale from 1 to 9 with corresponding reciprocal values (1/2 to 1/9). Radio buttons are used to select the desired value.

choose a value for: Cost effective to run in relation to Usuable

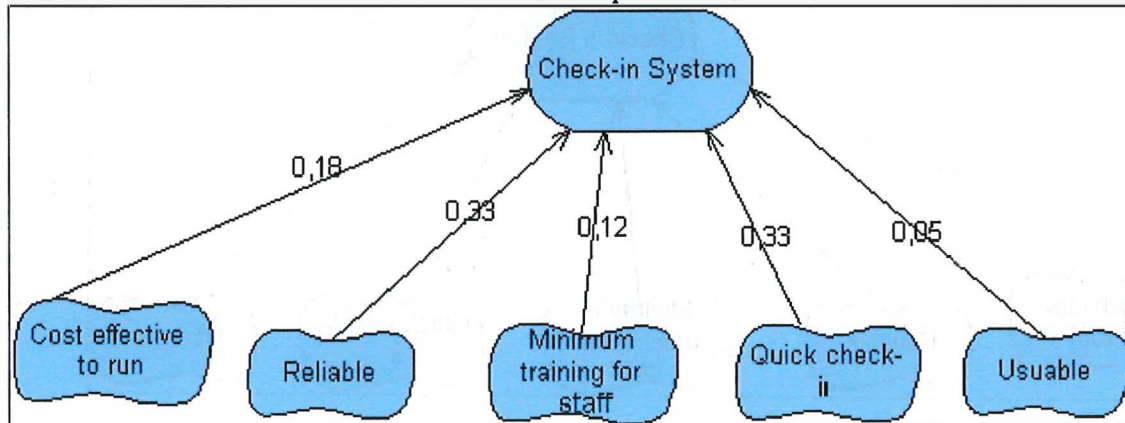
Equal importance Moderate importance Essential or strong importance Very strong importance Absolute importance Clear Cell

☒ 1
 ☐ 3 ☐ 1/3
 ☐ 5 ☐ 1/5
 ☐ 7 ☐ 1/7
 ☐ 9 ☐ 1/9
 ☐ Clear

☐ 2 ☐ 1/2
 ☐ 4 ☐ 1/4
 ☐ 6 ☐ 1/6
 ☐ 8 ☐ 1/8

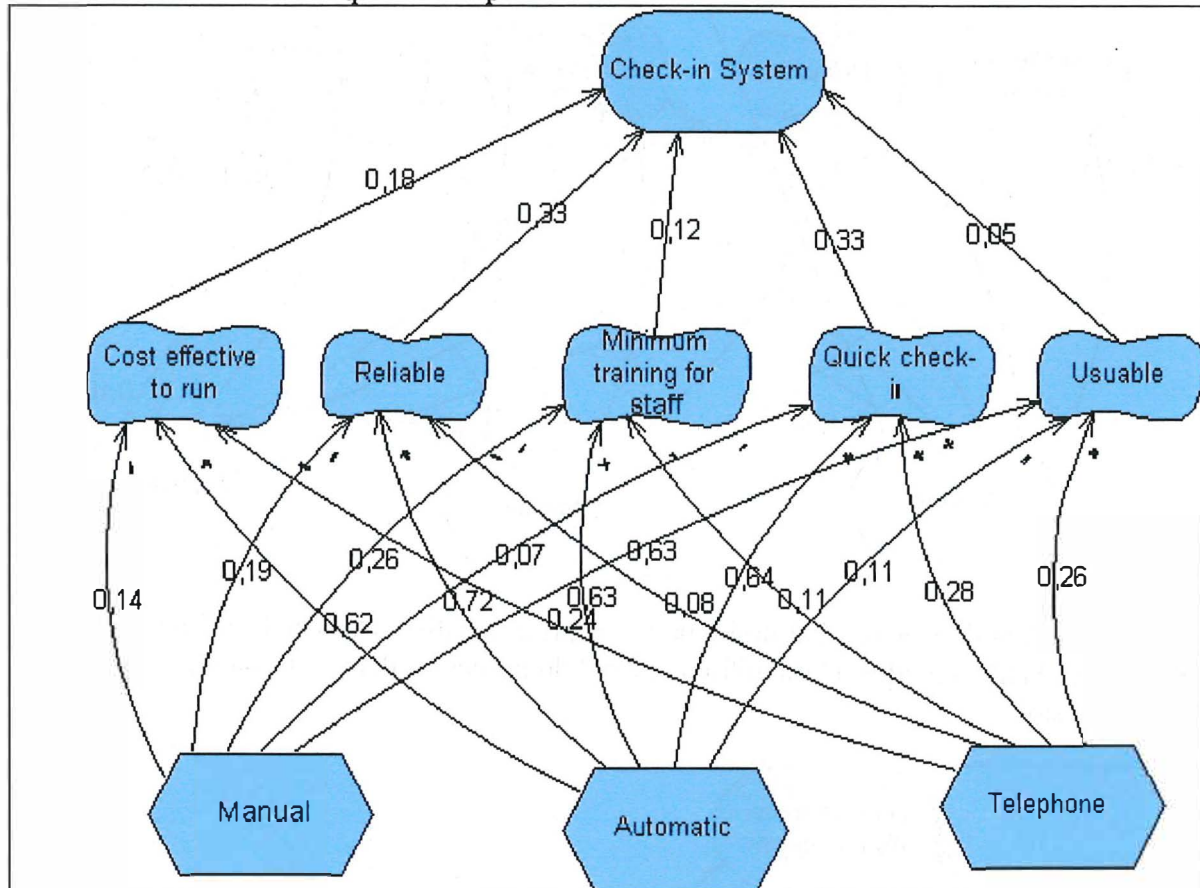
Buttons: Ok, Cancel

Une fois cela fait, Excel nous donne le poids relatif pour chaque critère. On peut donc importer ces valeurs dans le modèle Redepend qui va les placer sur les liens correspondants.



2^{ème} étape

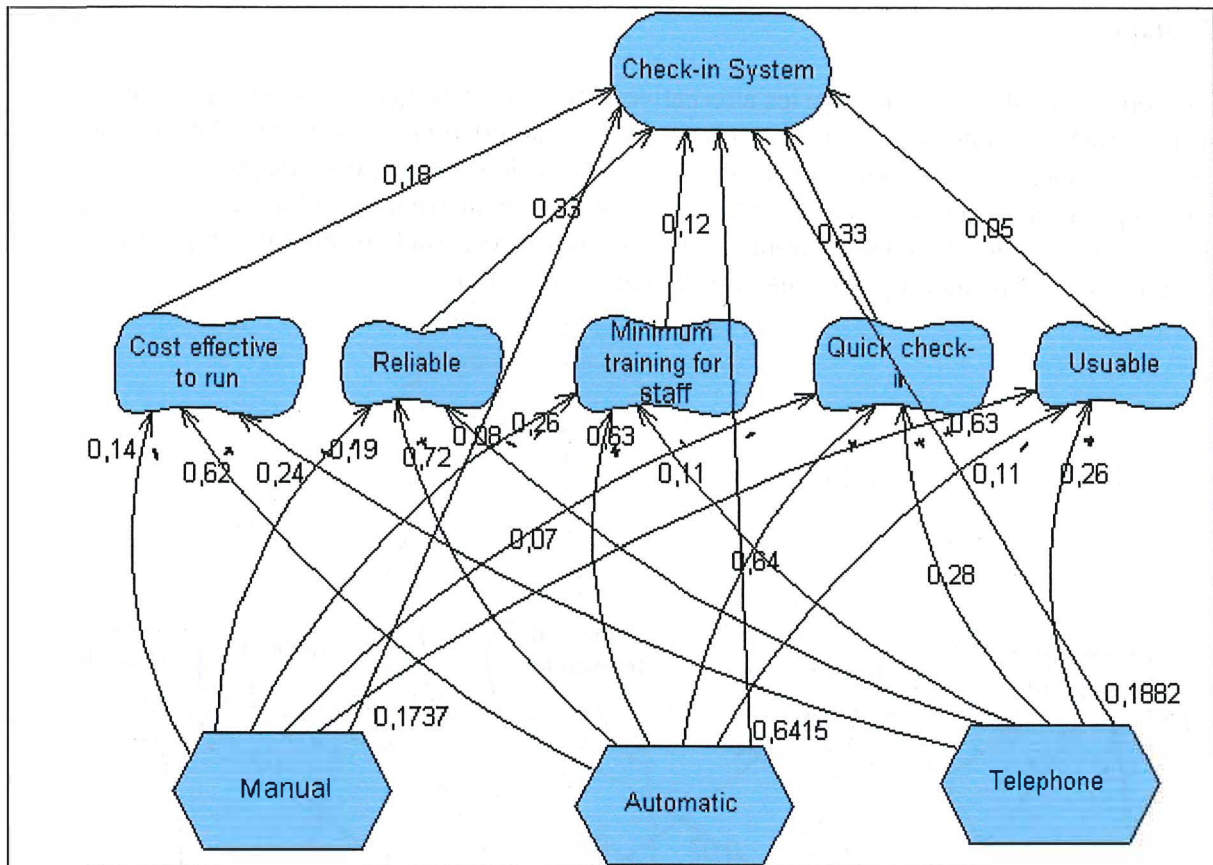
Dans cette étape il s'agit d'inclure les alternatives. Celles-ci sont généralement représentées par des « tasks » ou des « soft goals » suivant leurs caractéristiques respectives. On suit alors un raisonnement similaire pour construire les matrices alternatives pour chaque critère de décision par rapport aux alternatives. On relie donc chaque alternative à chaque critère. Une fois cela fait, on peut donc lancer pour chaque critère de décision la routine Excel pour le calcul matriciel. Elle identique à celle pour la matrice de critère.



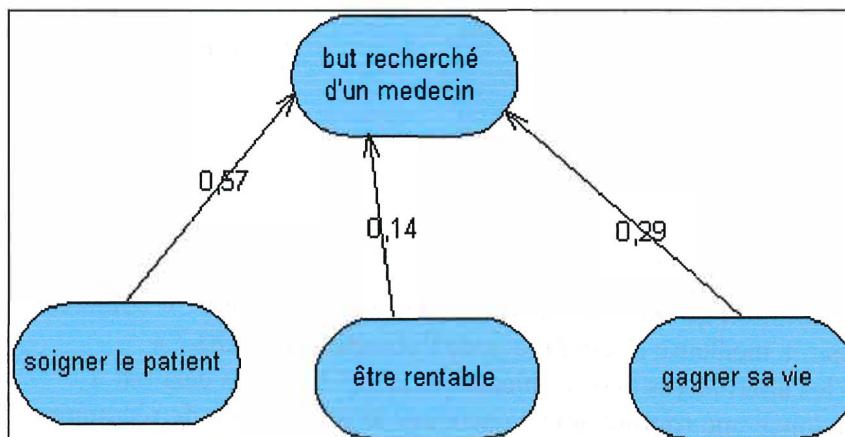
3^{ème} étape

La 2^{ème} étape étant réalisée, la question qu'on peut alors se poser est la suivante : « comment avec les valeurs trouvées choisir la meilleure alternative pour l'objectif recherché ? »

Pour réaliser cela il faut relier les alternatives avec l'objectif principal. Une fois cela fait à partir de l'objectif principal on lance une routine qui va parcourir le schéma afin de récupérer les valeurs déjà préalablement trouvées dans les étapes antérieures et avec celles-ci calculer la meilleure alternative possible en fonction des poids relatifs de chaque critère et alternative.



Au final on obtient donc le résultat désiré pour chaque alternative. Dans la deuxième application d'AHP au modèle i* on utilise l'étape 2 du processus défini plus haut pour obtenir le résultat désiré.

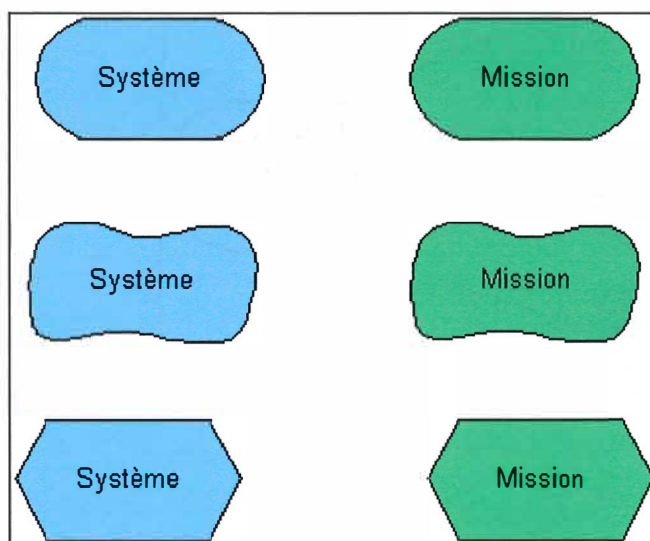


2 Addenda au chapitre 6

2.1 Buts de système et but de mission

La question qu'on pourrait se poser est de savoir quel est l'intérêt de distinguer les buts de mission et de système dans le modèle i*?

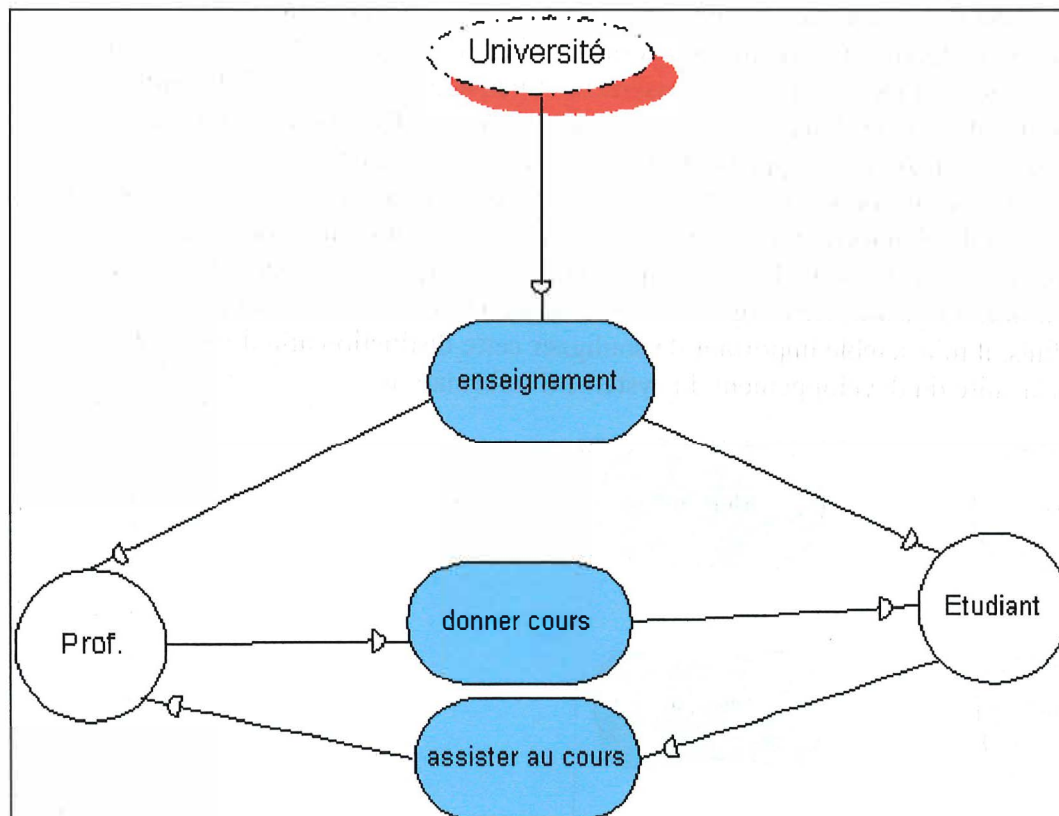
Le modèle i* rendant compte uniquement des dépendances entre acteurs et de la décomposition de chaque objectif recherché par les acteurs, il apparaît donc à première vue superflu de faire une telle distinction. Cependant comme on l'a vu au chapitre 2 les buts de mission sont des buts décrivent la vocation externe de l'institution. Ils sont la raison d'être de l'organisation. Ils ne sont pas définis rigoureusement et peuvent être sujets à différentes interprétations. Tandis que les buts liés au système sont ceux que l'institution poursuit pour assurer son existence et son développement. Ils concernent directement le fonctionnement de l'organisation et de son personnel. Ils sont donc plus concrets et mieux définis. C'est pourquoi il semble plus facile de bien formaliser un but de système qu'un but de mission étant donné sa définition. En effet vu que les buts de mission peuvent être interpréter de façon différente et que leur description est parfois très vague contrairement au but de système qui peuvent être clairement définis, il m'a semblé important de souligner cette distinction afin d'y prêter attention dans la suite du développement du système d'information.



2.2 Buts spécifiques et buts de l'organisation

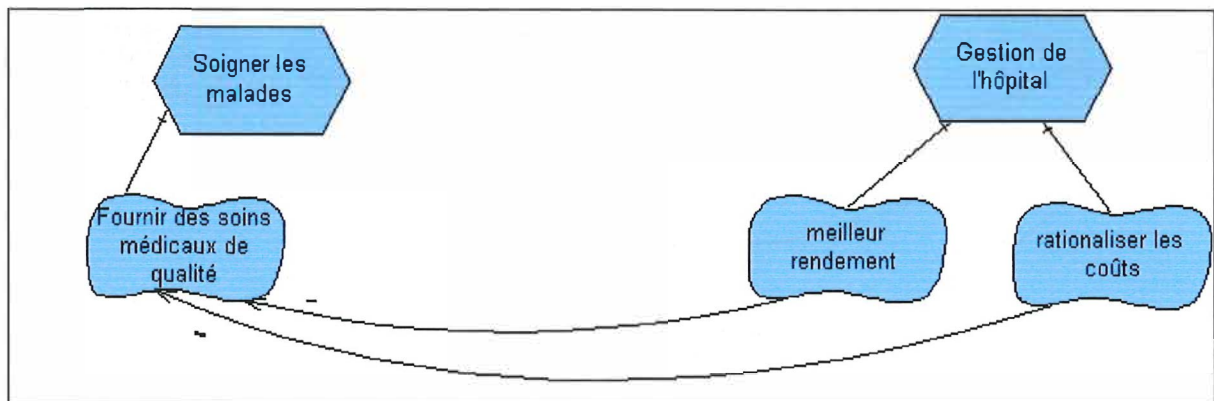
Comme c'est expliqué au chapitre 2, la différence entre les buts individuels et les buts de l'organisation est que ces derniers sont des buts collectifs poursuivis par l'organisation tandis que les buts individuels par les acteurs. Etant donné que le modèle i* est un modèle orienté acteur, il semble donc difficile parfois de représenter tous les buts de l'organisation uniquement en terme de dépendance entre acteur qui met en évidence les buts spécifiques de chacun de ces acteurs. Bien que la plupart du temps les buts de l'organisation peuvent être modélisés car ils coïncident avec les buts recherchés par chaque acteur, il arrive parfois qu'on

ne sache décrire les objectifs de l'organisation uniquement en terme de relations entre acteurs. C'est pourquoi il m'a paru judicieux d'introduire le concept d'organisation considéré comme un acteur à part entière de façon à permettre de traduire en terme i* tous les buts de l'organisation. De plus cela permet de mettre en évidence les buts spécifiques par rapport aux buts de mission de façon à montrer que les buts recherchés avec les acteurs concordent bien avec ceux de l'organisation et comment ces buts contribuent dans l'accomplissement des buts collectifs de l'organisation. Dans l'exemple qui suit, on peut voir qu'on aurait pu se passer de mettre l'université dans le schéma. Néanmoins cela permet de montrer que le but «enseignement» poursuivi par les deux acteurs est un but qui concorde bien avec la mission de l'université et que c'est un but recherché par celle-ci.

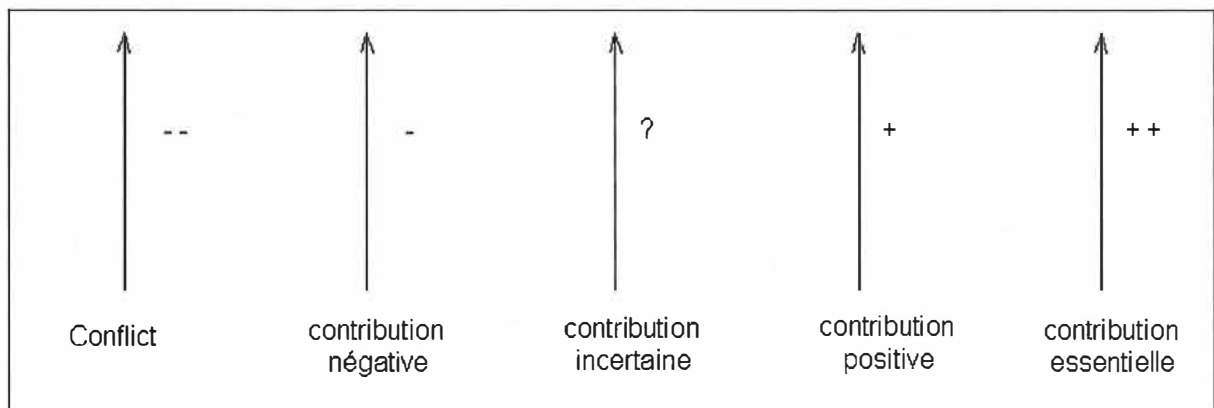


2.3 Les conflits entre buts

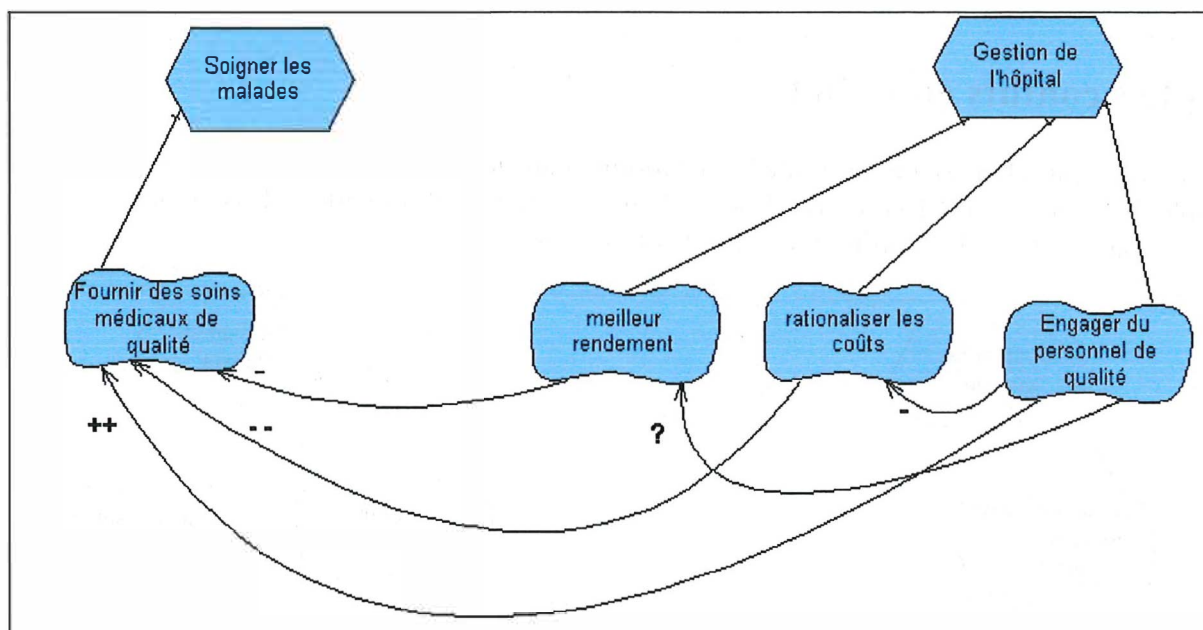
Les conflits entre but sont assez difficiles à exprimer dans le modèle i*. La façon la plus simple de le faire est de décomposer l'objectif en « soft goal » et d'exprimer les conflits entre ceux-ci par des liens de contributions positifs ou négatifs.



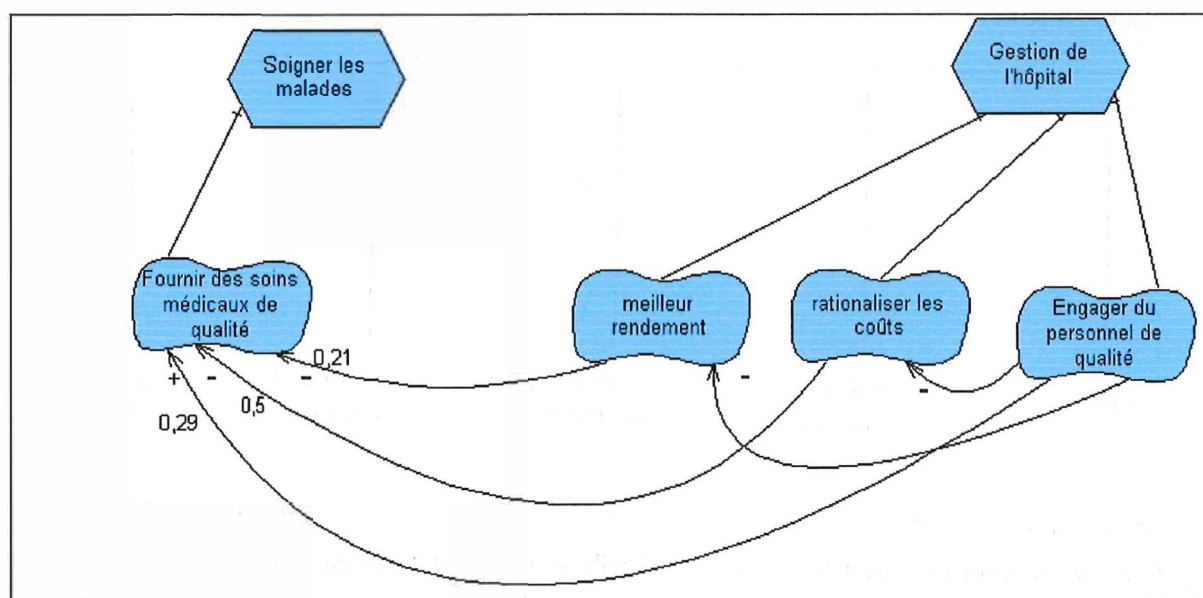
Cependant cette notation est assez limitée car cela permet seulement de montrer s'il y a une collaboration ou un conflit. Le niveau d'intensité de ces relations n'est pas défini. Une solution serait de prendre en compte serait donc d'utiliser différents liens de contribution issus du modèle NFR



Ces liens permettent de varier le niveau de contribution positif ou négatif d'un « soft goal » par rapport à un autre.



Une autre possibilité pour exprimer le degré de contribution ou de conflits serait d'utiliser la méthode AHP. En effet lorsque plusieurs « soft goals » contribuent positivement ou négativement à un objectif, il serait intéressant d'établir un classement entre ces différentes contributions de façon à permettre de mettre en évidence les « soft goals » les plus pertinents.



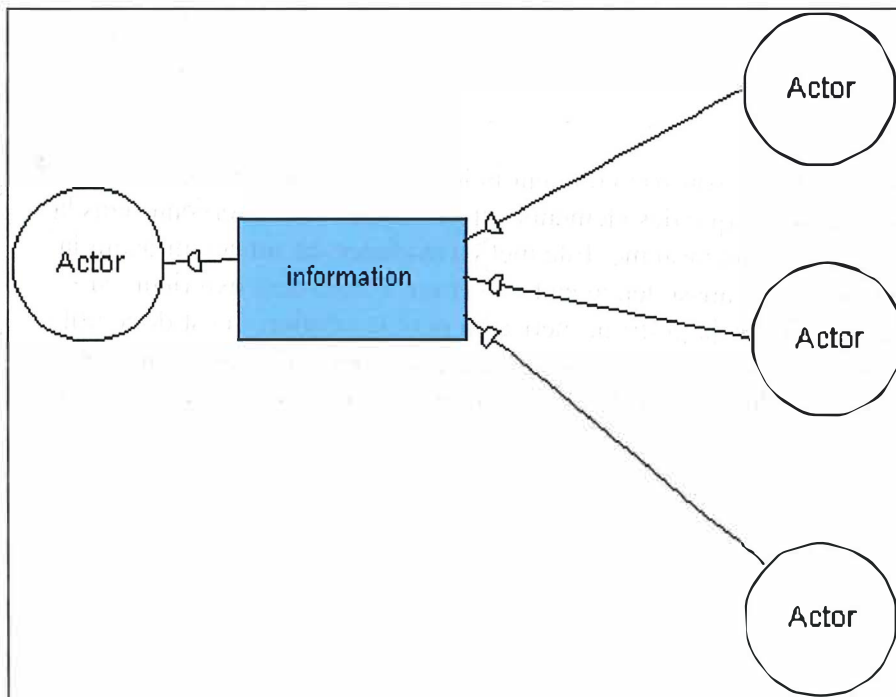
Une dernière solution aussi à envisager lorsqu'il n'y a qu'un « soft goal » contribuant à un autre serait d'ajouter une valeur (nombre) sur base d'une échelle au lien de façon à monter le degré de contribution.

De ces trois solutions la deuxième semble la meilleure car elle garde encore une certaine subjectivité du modèle. Cela correspond donc mieux à la « philosophie » initiale du modèle i^* . Bien que AHP s'avère être une bonne solution objective, le caractère trop restrictif de l'usage de valeurs numérique en fait une solution trop rigide. D'autant plus que ces valeurs peuvent varier selon la position et le point de vue de celui qui les calcule. Dans l'exemple ici cela a été réalisé sans préférence pour un parti particulier. Les résultats auraient été fortement

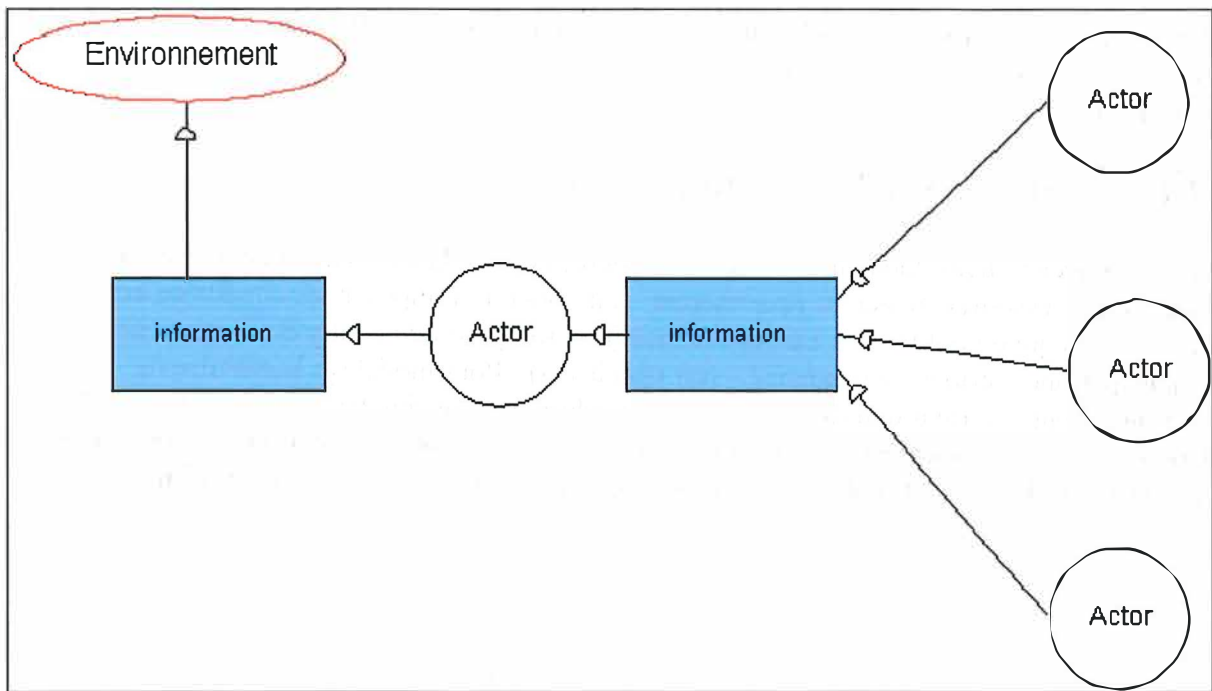
différents si c'était la direction ou les médecins qui auraient réalisé cette analyse. Enfin la dernière solution est aussi une solution trop catégorique qui a tendance à rendre le modèle trop objectif.

2.4 La relation avec l'environnement

La relation avec l'environnement peut être considérée comme la maîtrise d'une ressource. Grâce à cette ressource un acteur peut exercer son pouvoir et imposer ses conditions aux autres acteurs qui en ont besoin. La dépendance de cette ressource peut y être encore plus marquée par une « critical dependency » (cfr chapitre 3). Pour modéliser la maîtrise de l'environnement extérieur dans le modèle i* une solution serait simplement de considérer la relation avec l'environnement comme une ressource que détient un acteur dans la relation de dépendance. Cela reviendrait donc à la même chose que dans le cas du contrôle et l'usage de l'information.



Une modification ne s'avère donc pas absolument nécessaire. Cependant il est peut-être judicieux de bien mettre en évidence qu'un acteur a une maîtrise d'une ressource et que celui-ci cette maîtrise provient d'une ressource extérieure à l'organisation.



Bien que cette modification mineure ne soit pas vraiment indispensable, elle semble cependant intéressante afin de montrer que des éléments extérieurs peuvent intervenir dans la réalisation des objectifs des acteurs d'un système. Elle met en évidence qu'un acteur ayant la maîtrise d'une ressource extérieure est aussi dépendant de cet environnement extérieur. Si cette ressource viendrait à disparaître cela pourrait mettre en péril la relation. Il est donc utile, je pense, de parfois considérer certains éléments extérieurs dans le modèle i^* . Par contre cette modification a tendance à s'écarter du concept de i^* qui est un concept orienté acteurs et buts.

3. Annexe

Cette annexe inclut une explication détaillée de l'utilisation de la méthode AHP dans Redepend et une explication détaillée de l'algorithme AHP.

The AHP method and the i^* Model

1. The Analytic Hierarchy Process (AHP)

“It is better to be approximately right than precisely wrong», Thomas L. Saaty”

Managers are faced every day with complex decisions consisting of factors that are interconnected and dependent upon one another. It is almost impossible to make such decisions based on just personal intuition and thought. The Analytic Hierarchy Process (AHP) is a decision making process that provides a systematic method of considering all the elements of a problem. It organizes the problem into smaller parts and then only calls for simple pair wise judgments to develop a hierarchy. This hierarchy is then manipulated analytically to produce a final matrix representing the overall priorities of the alternatives relative to each other. One can then make a logical decision based on the pair wise comparisons made between the alternatives and the criteria being used in the decision.

The AHP allows managers to make simple comparisons of the factors involved in a decision, thus producing a hierarchy of the decision. The AHP provides managers with logical decisions based on analytical methods. This eliminates much of the chance that is often confronted with decision-making. The AHP also allows managers to consider both tangible and intangible factors when constructing the hierarchy. This ensures that the decision will be based on more than just financial or other measurable characteristics.

The Analytic Hierarchy Process (AHP) is then a powerful and flexible decision-making process to help people set priorities and make the best decision when both qualitative and quantitative aspects of a decision need to be considered.

2. Methodology of AHP

Step 1: Completely define problem and develop a hierarchy which will accurately represent the problem using the following Guidelines:

- Level 1- Final goal or objective
- Level 2- Criteria used to judge alternatives
- Level 3- Alternatives

The Level 1 consist of identify the general, final objective in a problem (Buying a house, chose a best machine a process of production for example). The second level is using to show criteria that compose the final objective like the time, the fare. The last level contains each alternative, possibilities for the level 1.

Develop matrices that compare the criteria with themselves (within level 2) and the alternatives with each criteria (between level 2 and level 3). For that, we use a scale of relative importance as follow:

Saaty's scale of relative importance

Intensity of relative importance	Definition	Explanation
1	Equal importance	2 activities contribute equally to the objective
3	Moderate importance	Experience and judgment slightly favour one over another
5	Essential or strong importance	Experience and judgment suggest a strong favour one over another
7	Very strong importance	An activity is strongly favoured and its dominance is demonstrated in practice
9	Absolute importance	Evidence favouring this activity is of absolute affirmation
2,4,6,8 Reciprocals	Intermediate value	If activity I has one of above numbers assigned when compared to J; then J has the reciprocal when compared to I

Step 3 : Compute priority weights of each matrix using eigenvalues. To this we must normalize the matrix(divide by the sums of the columns, and average across rows to get the relative weights).

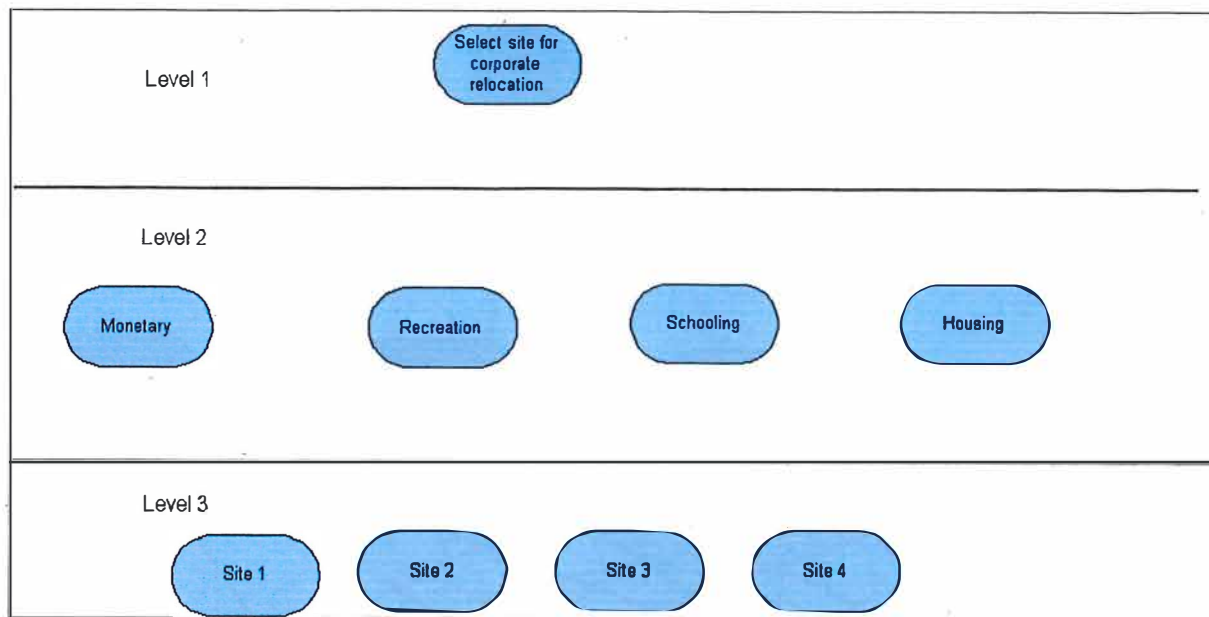
Step 4 : Compute composite priorities of alternatives by linearly adding the priority weights of the corresponding criteria in *criteria matrix* with the weights for these criteria in *alternative matrix* corresponding with the alternative.

2.1. Example : Select a site for corporate relocation

An industry must choose a site for a corporate relocation. It has the choice between 4 sites. The attributes for the corporate are Monetary, Recreation, Schooling and Housing. In this example we will see which is the best choice.

The first step consists of developing a hierarchy following the 3 levels. Here is the diagram representing the first step¹:

¹ This not a i* model



After building this drawing we can deal with the second step. In this step we must develop a matrix that compare the criteria with themselves (within Level 2) and alternatives with each criteria(between Level 2 and 3)

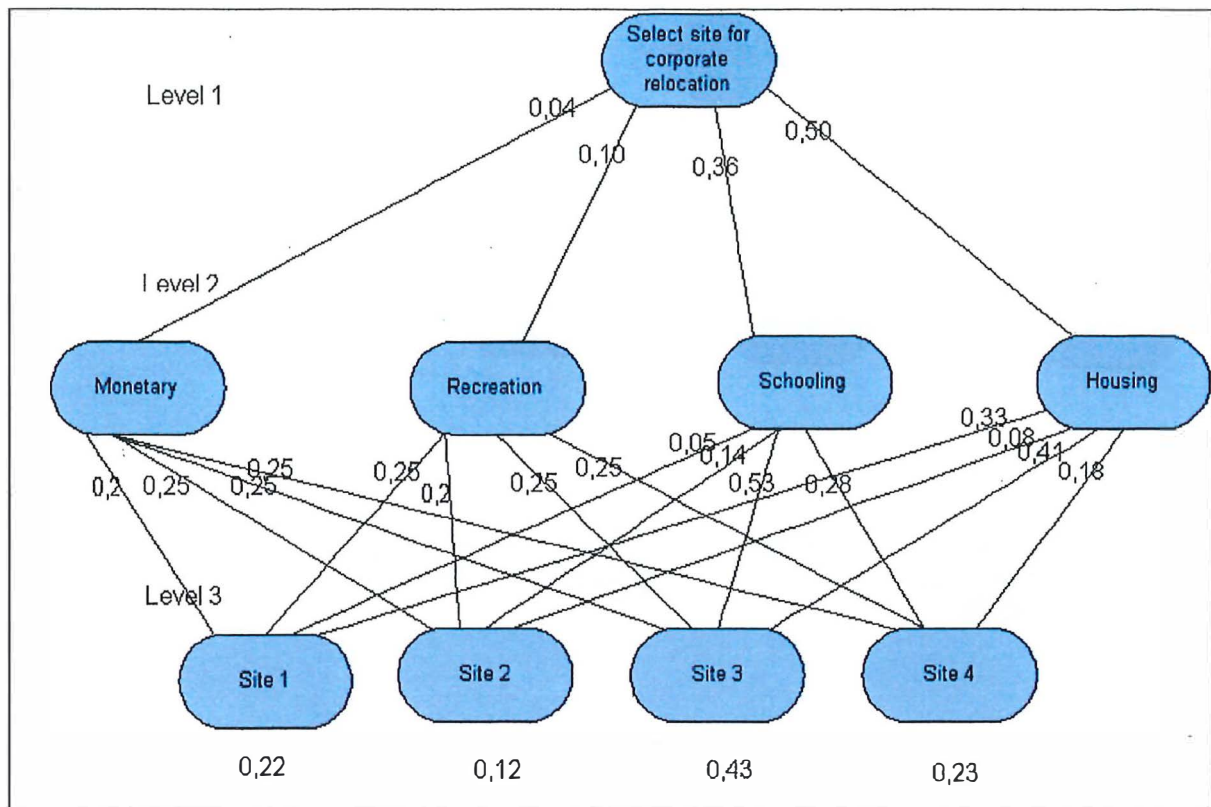
	Monetary	Recreation	Schooling	Housing	Weight
Monetary	1	0,33	0,11	0,11	0,04
Recreation	3	1	0,2	0,2	0,1
Schooling	9	5	1	0,5	0,36
Housing	9	5	2	1	0,5

The criteria Matrix

Here is for example the Matrix For Schooling in relation with the 4 sites:

	Site 1	Site 2	Site 3	Site 4	Weight
Site 1	1	0,2	0,11	0,2	0,05
Site 2	5	1	0,2	0,33	0,14
Site 3	9	5	1	8	0,53
Site 4	5	3	0,13	1	0,28

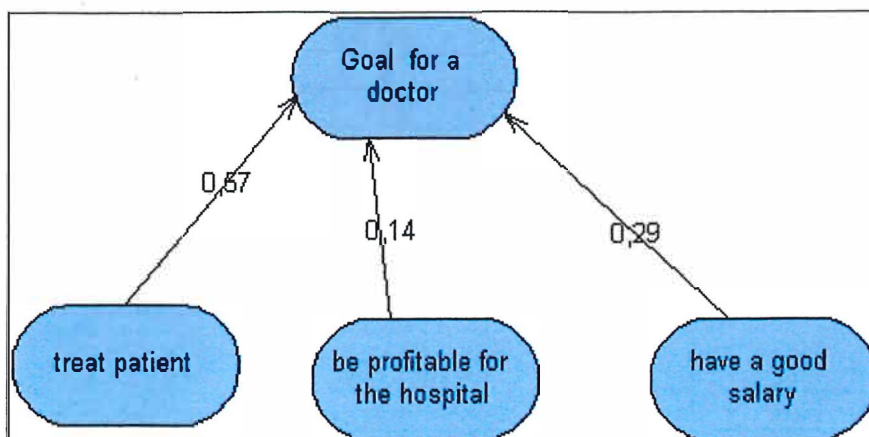
After building alternative matrix we can go to the next step. The next consist of to compute priority weights of each matrix(you can see this in the row called Weight). In the final step we must simply the weight of the attribute from step 1 with the weight of alternative with respect to that attribute (from step 2). And we have a diagram like this:



3. What's the possible link between i^* and AHP ?

The problem in i^* is that when we have a goal (or soft-goal) decomposition into goals (soft-goals or tasks) by means end link or a contribute link, we can explain the relative importance of these goals in this decomposition. To have an absolute objective value, we can use AHP to do that and particularly the step 1 and 2 of this process. We can consider as level 1 in an i^* diagram a goal or sub goal, and as level 2 their sub goal. Then, we can build the *criteria matrix* with these ones follows the Saaty's scale. With this method you can directly see what is the most important sub goal. For example if you have a simple problem like this :

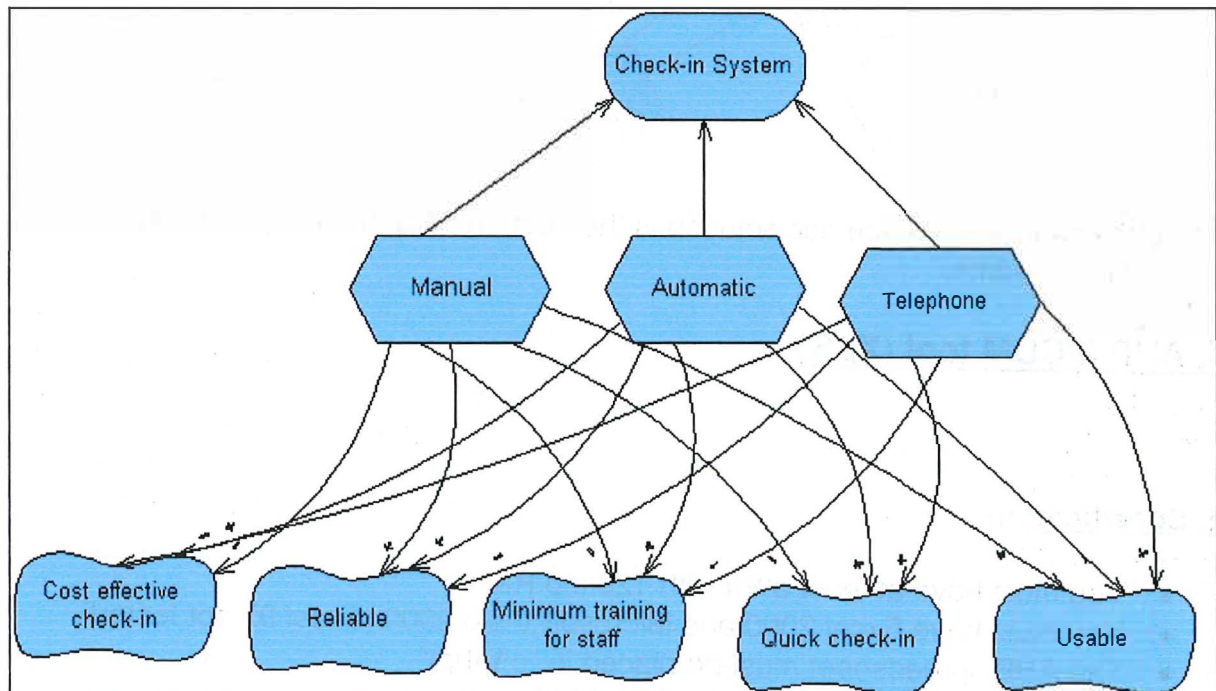
A doctor can have three objectives. The first is, of course treat the patients, the second to be rentable and the last like everybody to have a good salary for living. This problem can be described with i^* like this:



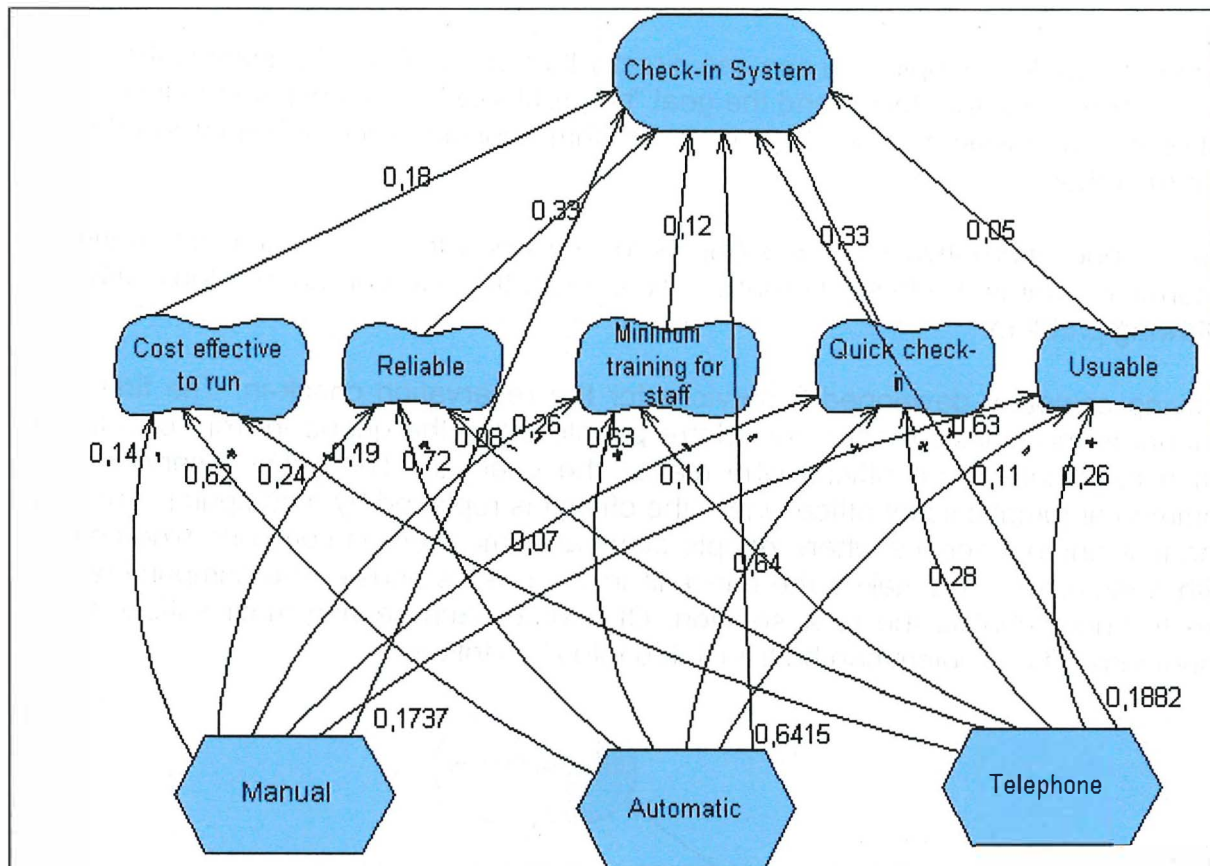
With this simple example you can see directly that the goal "treat patient is the more important goal for the doctor and the goal "be profitable for the hospital the less. It is better that a classic sort because the algorithm consider each sub goal in relation with the other.

The second alternative for using Ahp is to see when there are some alternatives, determine what is the best alternative. To explain this, we can have a look with the following problem:

A flying company developed 3 services for the reservation check-in. The first, the manual is the classic reservation where people make the queue in front of a ticket office and there is an official who makes the check-in. The second service is a common automatic ticket office where the official is replaced by a computer. The last one is a original service where people stays at home there is someone who comes with a motorbike and makes the check-in in the client's house. The company would like to know what is the best solution. Of course each service must satisfy some conditions. The problem can be translated into i^* as follow:



But the problem with this diagram is that you are not able to determine what is the best solution. So with the integration of AHP method you can calculate the best solution.



You can see that the automatic solution is the best and that the telephone and the manual is the same.

4. AHP MCDM tool (XLS)

1. Specification

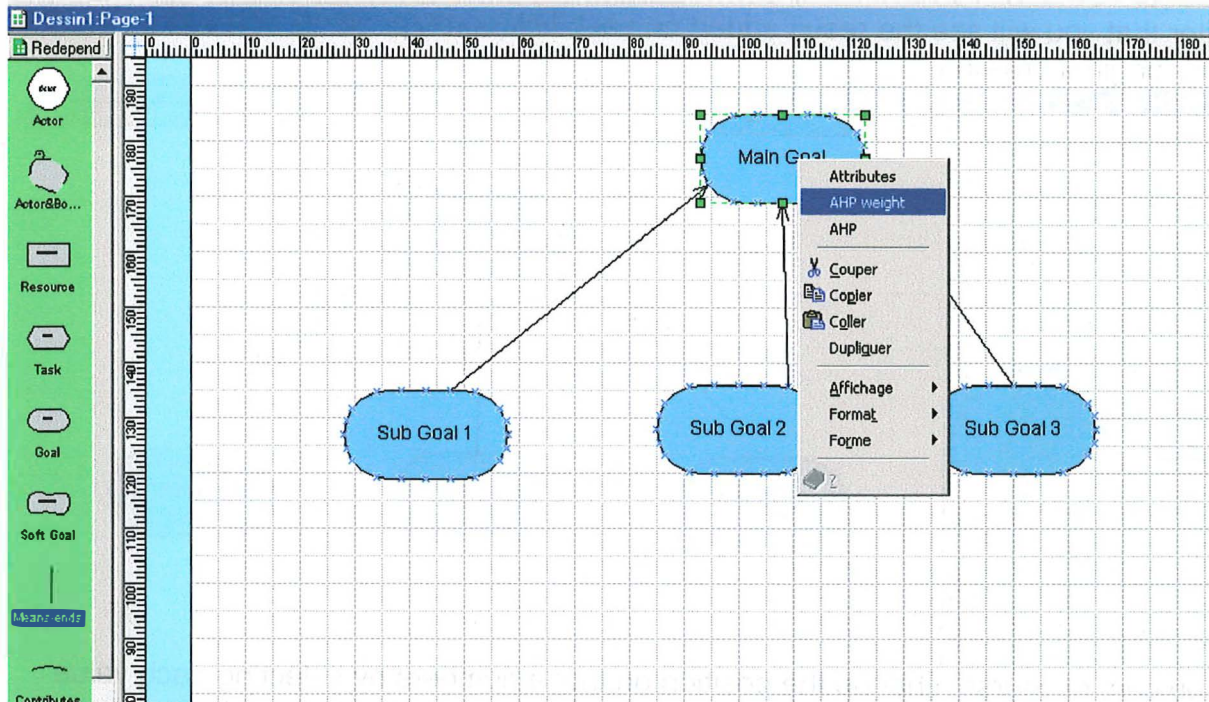
- You must have at least a PII with 128 mb Ram.
- You must have Excel 2000 and Microsoft Visio 2000 (Excel 97 not tested).
- The AHP spreadsheet must be placed in "c:\Ahp\"
- The screen resolution must be 1024 * 768.
- The connections check-in must be enabled in your drawing.
- All the goal softgoal must have a name and the name must be having a different name. The name must be also different that the shape name. The name can be greatest that 255 characters

2. Tutorial

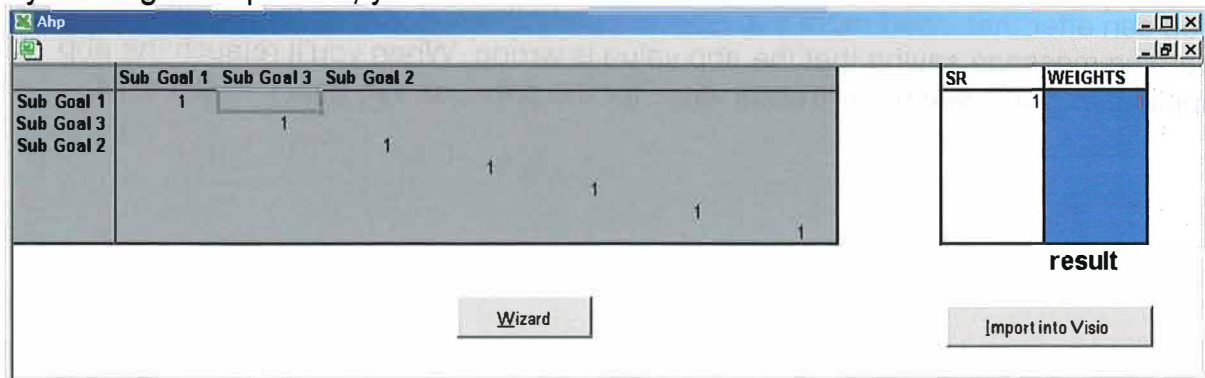
2.1. Weight for a goal or softgoal

The AHP MCDM tool can work with Goal or Soft Goal. If you a have one "goal" (or "soft goal") linked with a minimum 2 and maximum 7 "means-ends"("contributes

links”) to another “goals” (or soft goals) you can run the AHP Tool by a right click on this “goal”.



By clicking on ahp menu, you will have a window like this :

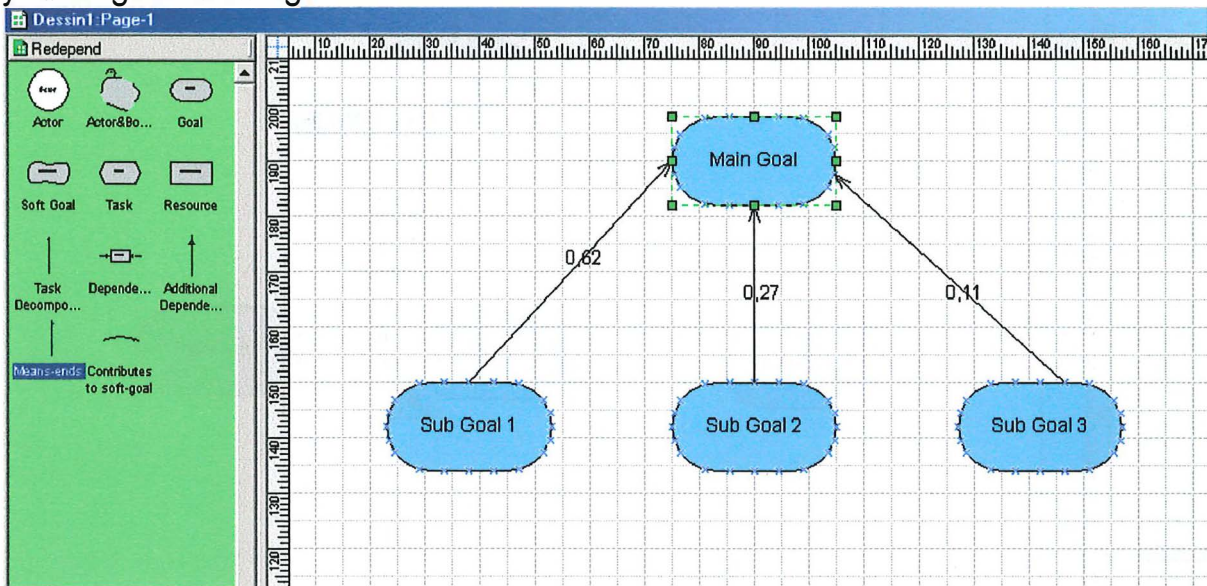


A matrix including names of the sub goal, a sr column, a weights column and 2 buttons composes this window. By clicking on the “wizard” button you’ll fill the matrix with ahp values. To do this, a window with radio button for each cell of the matrix can help you.

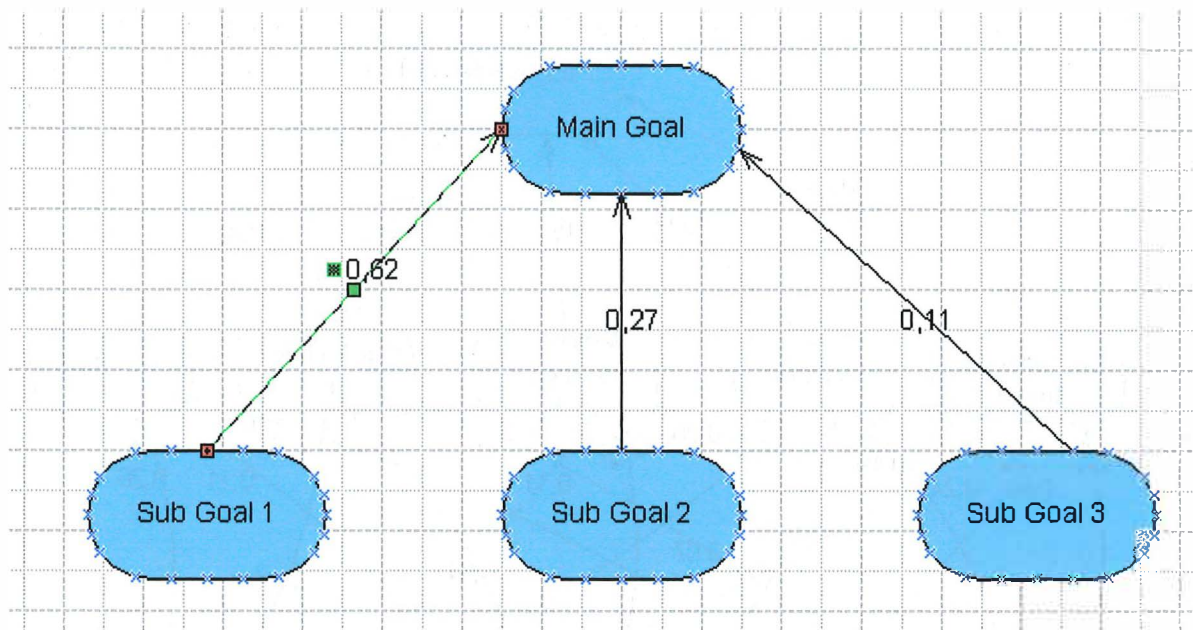
You can also fill the matrix manually by clicking on cell (The windows above appears) Remember that you can only fill the cell corresponding with the name of sub goal. If you fill another cells, the result will be wrong.

After filling the entire cell you can import the result showed in the column weights into visio by clicking on the corresponding button. The sr column is only for intermediate values. You don't have to worry about it.

After that you will see the result added on each link (corresponding with sub goal) on your original drawing.



You can, of course, change the position on these numbers by selecting each value and pick up with mouse the small green square associated with the selected number. You can after that insert more sub goal or delete them. If you do that you'll have a warning message saying that the ahp value is wrong. When you'll relaunch the ahp application you'll see the previous value for the previous sub goal that still exist.

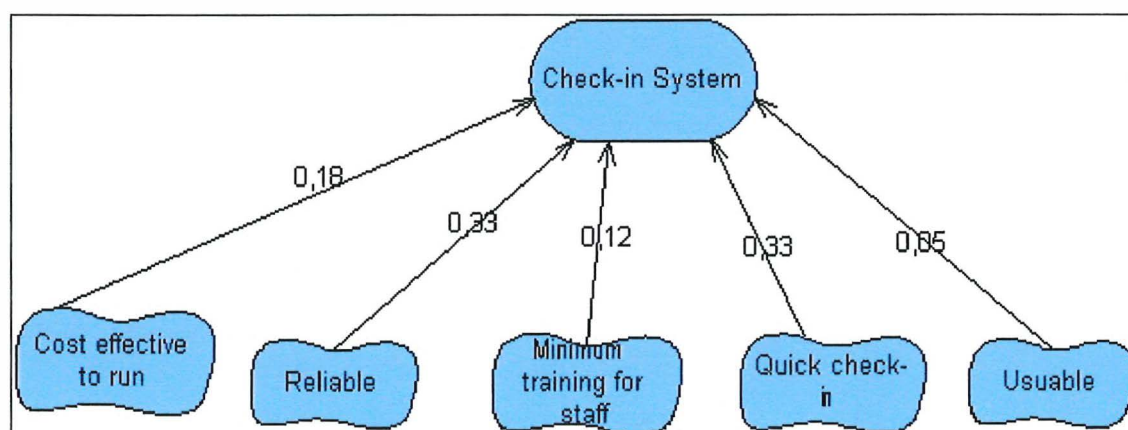


2.2. AHP to determine the best alternative

For example with the airport check-in, if you want to determine what is the best alternative you must build a diagram like this:

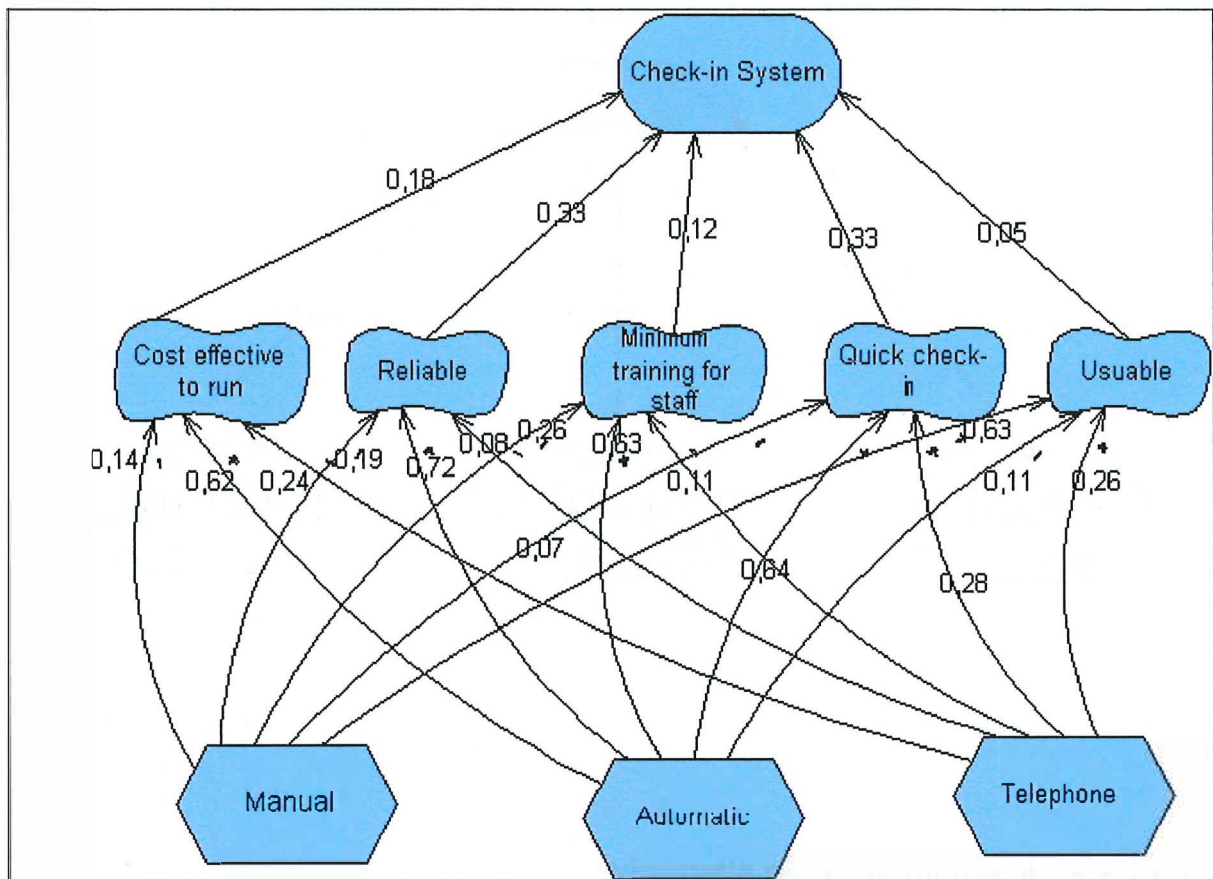
2.2.1. First step

Build a diagram with the final objective and link it with the criteria and calculate the weight for this objective(criteria matrix):



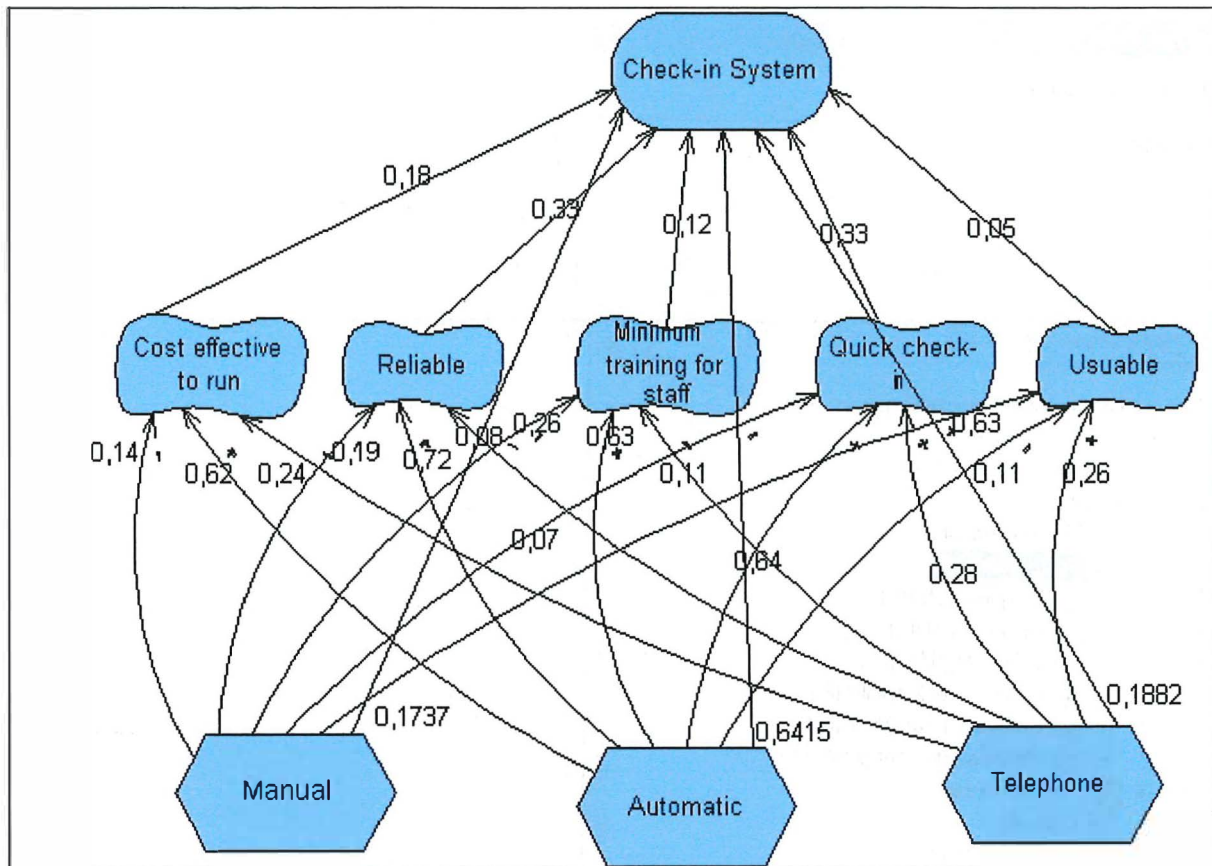
2.2.2. Second step

Put on the diagram the alternatives and link it with the criteria. Calculate the weight for each criteria(alternative matrix):



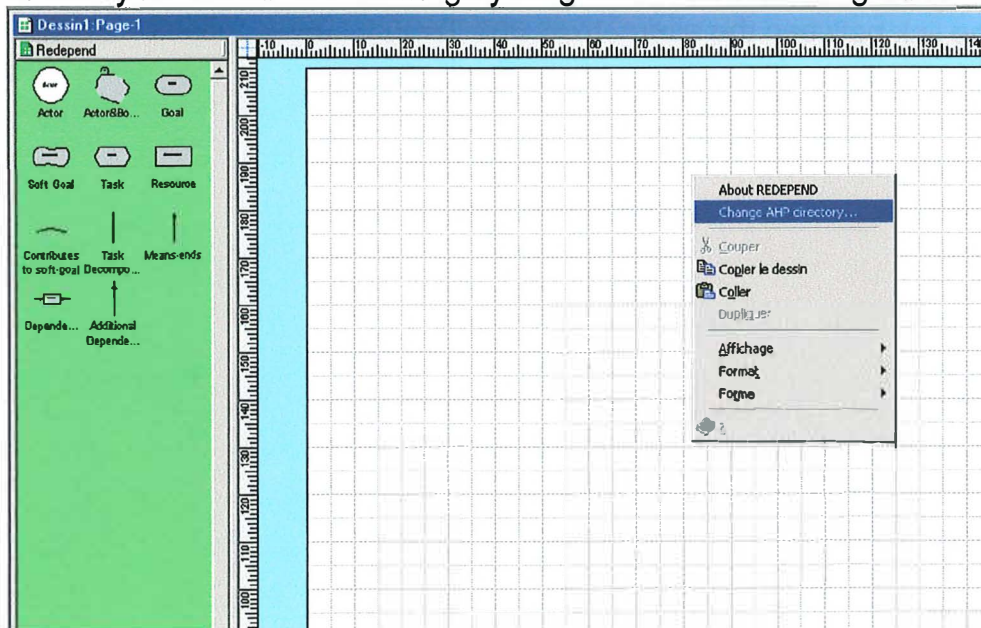
2.2.3. Third Step

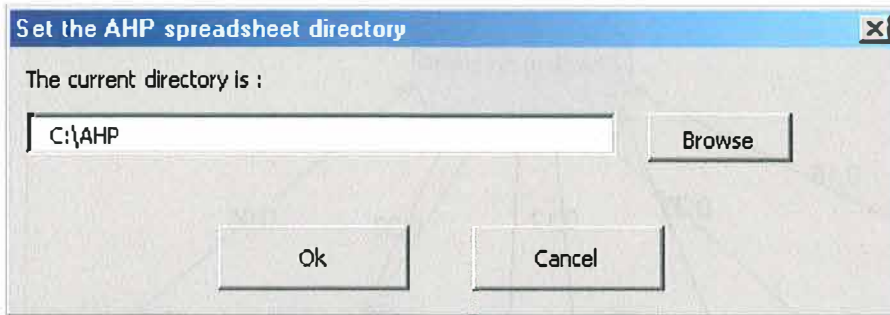
Link the alternatives with the main goal and calculate the priority vector for these by clicking on the main goal and selecting "AHP" in the menu:



2.3. Additional functions

By default, the spreadsheet is placed in "c:\ahp\" but if you want you can change this directory for the current drawing by a right click on the background.





3. Known bugs

If you want to modify the Ahp weight between the criteria and the main goal, please disconnect first the link between the alternatives and the main goal.

Appendix: Mathematical explanation of AHP

This is a mathematical explanation of AHP. I illustred it with an example for a better understanding.

One Thursday morning, Charles, instead of attending his Management Science Techniques for Consultants class, was mulling over his four job offers. His offers came from Acme Manufacturing, Bankers Bank, Creative Consulting, and Dynamic Decision Making. He knew that factors such as location, salary, amount of management science (which he loved), and long term prospects were important to him, but he wanted some way to formalize the relative importance, and some way to evaluate each job offer. Luckily, he attended the following Tuesday class of MSTC, who showed him one way to think about these problems. This technique is called the Analytic Hierarchy Process.

The first step in AHP is to ignore the jobs and just decide the relative importance of the objectives. Charles does this by comparing each pair of objectives and ranking them on the following scale: Comparing objective i and objective j (where i is assumed to be at least as important as j), give a value a_{ij} as follows:

- 1 Objectives i and j are of equal importance
- 3 Objective i is weakly more important than j
- 5 Objective i is strongly more important than j
- 7 Objective i is very strongly more important than j
- 9 Objective i is absolutely more important than j
- 2,4,6,8 Intermediate values

Table 1: Pairwise comparison values

Of course, we set $a_{ii} = 1$. Furthermore, if we set $a_{ij} = k$, then we set $a_{ji} = \frac{1}{k}$. Charles, thinking hard about his preferences, comes up with the following table:

	Location	Salary	MS	Long
Location	1	1/5	1/3	1/2
Salary	5	1	2	4
MS	3	1/2	1	3
Long	2	1/4	1/3	1

Table 2: Preferences on Objectives

Now, the AHP is going to make some simple calculations to determine the overall weight that Charles is assigning to each objective: this weight will be between 0 and 1, and the total weights will add up to 1. We do that by taking each entry and dividing by the sum of the column it appears in. For instance the (Location,Location)

entry would end up as $\frac{1}{1+5+3+2} = .091$. The other entries become:

	Location	Salary	MS	Long	Average
Location	.091	.102	.091	.059	.086
Salary	.455	.513	.545	.471	.496
MS	.273	.256	.273	.353	.289
Long	.182	.128	.091	.118	.130

Table 3: Weights on Objectives

This suggests that about half of my objective weight is on salary, 30% on amount of management science, 13% on long term prospects, and 9% on location.

Now, why does this magical transformation make sense? If we read down the first column in the original matrix, we have the values of each of the objectives, normalized by setting the value of location to be 1. Similarly, the second column are the values, normalizing with salary equals 1. For a perfectly consistent decision maker, each column should be identical, except for the normalization. By dividing by the total in each column, therefore, we would expect identical columns, with each entry giving the relative weight of the row's objective. By averaging across each row, we correct for any small inconsistencies in the decision making process.

Our next step is to evaluate all the jobs on each objective. For instance, if we take Location, if we prefer to be in the northeast (and preferably Boston), and the jobs are located in Pittsburgh, New York, Boston, and San Francisco respectively, then we might get the following matrix:

	A	B	C	D
A	1	1/2	1/3	5
B	2	1	1/2	7
C	3	2	1	9
D	1/5	1/7	1/9	1

Table 4: Location Scores

Again we can normalize (divide by the sums of the columns, and average across rows to get the relative weights of each job with regards to location. In this case, we get the following:

	A	B	C	D	Average
A	.161	.137	.171	.227	.174
B	.322	.275	.257	.312	.293
C	.484	.549	.514	.409	.489
D	.032	.040	.057	.045	.044

Table 5: Relative Location Scores

In words, of the the total ``Location Value" available, Job C and about 50%, B has about 30%, A has about 17% and D has about 4%. We can go through a similar process with Salary, amount of MS, and long term prospects. Suppose the relative values for the objectives can be given as follows:

	A	B	C	D
Location	.174	.293	.489	.044
Salary	.050	.444	.312	.194
MS	.210	.038	.354	.398
Long	.510	.012	.290	.188

Table 6: Relative scores for each objective

Recalling our overall weights, we can now get a value for each job. The value for Acme Manufacturing is

$$(.174)(.086) + (.050)(.496) + (.210)(.289) + (.510)(.130) = .164$$

Similarly, the value Bankers Bank is

$$(.293)(.086) + (.444)(.496) + (.038)(.289) + (.012)(.130) = .256$$

The value for Creative Consultants is .335, and that for Dynamic Decision is .238. Creative Consultants it is! Charles immediately makes his decision in order to have extra time to study for the MSTC exam.



0 1 0 0 7 6 8 5 1

*FM B16/2002/09/2

6. References

[1] Thomas Saaty. *The Analytic Hierarchy Process* . McGraw-Hill Co., 1980