

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Analyse de la compréhension des patrons de conception: expérimentation avec différentes représentations

Christophe, J; Soyeur, N

Award date:
2012

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

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

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

Take down policy

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

**Analyse de la compréhension des
 patrons de conception : expérimentation
 avec différentes représentations**

CHRISTOPHE Julien

SOYEUR Nicolas

Résumé

L'activité de maintenance est l'activité la plus coûteuse de tout projet logiciel. En réalité, une tâche préliminaire de compréhension logicielle est nécessaire à n'importe quelle activité de maintenance d'un logiciel. Des approches, telle que l'utilisation de diagrammes, ont été proposées afin d'offrir une vue des artefacts logiciels de plus haut niveau et ainsi faciliter la compréhension logicielle. D'un autre côté, les patrons de conception sont couramment utilisés au sein des architectures logicielles. En effet, ceux-ci ont pour avantage de faciliter la compréhension logicielle et donc l'activité de maintenance. Suite à ces constatations, certaines techniques de représentations des patrons de conception au sein des diagrammes de classes ont vu le jour. En 2008, Cepeda Porras [Por08] a mis en place une étude empirique évaluant l'impact de différentes représentations sur la compréhension des patrons de conception. Dans son étude, Cepeda Porras détecte des différences significatives pour trois différentes tâches de compréhension de patrons de conception. L'analyse de cette étude nous a permis de concevoir une nouvelle représentation, nommée « stereotype pattern enhanced UML diagrams » (SPE) et basée sur l'utilisation de couleurs et de stéréotypes UML. Au travers d'une expérience où nous utilisons un oculomètre, nous avons tenté de reproduire l'expérience de Cepeda Porras, tout en intégrant des éléments additionnels. Cette expérience est notamment l'occasion d'évaluer la représentation SPE. L'analyse des résultats illustre, à nouveau, des différences significatives pour les trois types de tâches. Par ailleurs, l'évaluation de la représentation SPE démontre une difficulté de différenciation des couleurs utilisées. Les améliorations envisageables sont donc un choix de couleurs suivant une métrique bien précise et une augmentation de la taille des traits de couleurs. En outre, nous présentons une amélioration d'une métrique utilisée dans le domaine de l'oculométrie, nommée distance relative entre chemins visuels. Enfin, nous avons apporté certaines contributions au logiciel d'analyse des données oculométriques (TAUPE).

Mots clés : génie logiciel empirique, compréhension logicielle, oculométrie, représentation de patrons de conception, mesure de l'effort.

Abstract

The maintenance activity is the most expensive of every software project. In fact, a preliminary task of software comprehension is necessary to any program maintenance activities. Some approaches, such as the use of diagrams, have been proposed in order to offer a more high-level view of software artifacts and thus to facilitate program comprehension. Moreover, design patterns are commonly used within software architectures. In fact, design patterns have the advantage of facilitating program comprehension and thus the maintenance activity. Subsequent to those findings, some representation techniques of design patterns in class diagrams have emerged. In 2008, Cepeda Porras [Por08] realised an empirical study assessing the impact of different representations on the comprehension of design patterns. In his study, Cepeda Porras detects significant differences for three different tasks of design pattern comprehension. The analysis of this study has allowed us to create a new representation, named "stereotype pattern enhanced" (SPE) and based on the use of colors and UML stereotypes. By means of an experiment where we use an eye tracker, we tried to replicate the experience of Cepeda Porras, while integrating additional elements. This experiment is especially an opportunity to evaluate the SPE representation. The analysis of the results demonstrates, again, significant differences for the three tasks. Furthermore, the evaluation of the SPE representation reveals a difficulty to distinguish the colors used. Possible improvements are a choice of colors following a precise metric and an increasing of the size of the colored lines. Moreover, we introduced an improvement of the metric relative distance between visual paths. Finally, we have contributed to the analysis software of eye tracking data (TAUPE).

Keywords : empirical software engineering, software comprehension, eye tracking, representation of design patterns, effort assessment.

Nous tenons à témoigner de toute notre gratitude à l'égard de différentes personnes qui ont concouru de près ou de loin à l'élaboration de ce mémoire.

Les premiers remerciements s'adressent à notre promoteur de stage, Monsieur Najj Habra, pour son engagement et son soutien qui nous ont notamment permis de réaliser notre stage à l'École Polytechnique de Montréal. Nous tenons également à remercier notre maître de stage, Monsieur Yann-Gaël Guéhéneuc pour sa disponibilité et sa gentillesse tout au long de notre stage.

Nous tenons ensuite à remercier vivement toutes les personnes qui nous ont aidés directement ou indirectement à réaliser ce mémoire, sans qui ce dernier n'aurait pas pu aboutir. A cet égard, nous pensons aux membres des laboratoires Ptidej Lab et SoccerLab qui ont accepté de participer à notre expérience. Nous remercions plus particulièrement Nasir Ali, Zohreh Sharafi et Zéphyrin Soh pour leur aide apportée quant à l'utilisation des appareils oculométriques disponibles au sein de l'École Polytechnique.

Nous adressons également de vifs remerciements à Madame Caroline Jennes et Madame Rita Celot pour leurs relectures attentives.

Enfin, nous remercions notre entourage qui nous a soutenu tout au long de ce mémoire. Nous pensons particulièrement à nos familles ainsi qu'à nos compagnes pour leurs encouragements de tous les jours.

Glossaire

Approche GQM Goal-Question-Metric définit une approche de mesure à trois niveaux pour les métriques logicielles. Les trois niveaux sont le niveau conceptuel (le but), le niveau opérationnel (la question) et le niveau quantitatif (la métrique) [BCR94].

Architecture logicielle L'architecture logicielle d'un programme ou d'un système informatique est la structure ou les structures du système, qui comprend les composants logiciels, les propriétés extérieurement visibles de ces composants, et les relations entre eux [DPA+92].

Charge mentale de travail La charge mentale de travail est définie comme la partie de la capacité de traitement ou des ressources nécessaires pour répondre aux exigences d'un système [Cai07].

Diagramme de classes Les diagrammes de classes UML sont la base de la modélisation orientée objet. Les modèles de classes montrent les classes dans un programme, leurs inter-relations (en incluant l'héritage, l'agrégation et l'association) et les opérations et les attributs des classes [Amb04].

Distance de Levenshtein La distance de Levenshtein représente la distance existante entre deux chaînes de caractères.

Dynamique Time Warping Dynamique Time Warping (DTW) est un système de mesure de distance entre deux courbes.

Enveloppe convexe L'enveloppe convexe d'un objet ou d'un regroupement d'objets géométriques est l'ensemble convexe le plus petit parmi ceux qui le contiennent.

Fixation Une fixation est une position du regard durant un parcours visuel. Chaque fixation possède des coordonnées (x, y) et un temps de début et de fin.

Fovéa La fovéa constitue la zone de meilleure acuité visuelle de l'être humain. Elle est située au centre de la rétine [Pal99].

Framework logiciel (ou cadriciel) Un *kit* de composants logiciels structurels qui servent à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel [wiki12].

Génie logiciel Le génie logiciel est l'application d'une approche systématique, disciplinée et quantifiable pour le développement, l'opération et la maintenance de logiciels [IEE90].

Identificateur Symbole formé d'une suite de caractères alphanumériques, employé en programmation informatique pour désigner ou utiliser une variable ou une fonction. [lar12].

Idiome Patron de bas niveau spécifique à un langage de programmation. Il décrit comment implémenter les aspects des composants ou des relations entre eux en utilisant les caractéristiques d'un langage donné [BMR+96].

Maintenabilité La facilité avec laquelle un système logiciel ou un composant peut être maintenu.

Oculomètre Un oculomètre (ou eye-tracker) est un appareil qui analyse des images de l'oeil humain. Ces images sont capturées par une caméra, souvent en lumière infrarouge, et permettent de calculer la direction du regard d'une personne.

Patron de conception Un patron de conception fournit un schéma pour affiner les sous-systèmes, les composants d'un système software, ou les relations existantes entre eux. Il décrit une structure récurrente de composants communicants qui résout un problème général de design dans un contexte particulier [GHJ+94].

Résultat significatif En statistiques, un résultat est dit statistiquement significatif lorsqu'il est improbable qu'il puisse être obtenu par un simple hasard.

Saccade La saccade est un mouvement rapide de l'œil qui se produit entre deux fixations.

Sujet Un sujet est une personne ayant passé notre expérience.

Surveillance de l'expérience Dans le cadre de notre expérience, cette activité consiste à surveiller un sujet, l'accompagner tout au long de l'expérience mais également à vérifier que l'enregistrement oculométrique ait bien lieu.

Zone d'intérêt (AOI) Une zone d'intérêt représente une zone d'une image d'un diagramme. Chaque zone possède un niveau d'intérêt : pertinente, non-pertinente ou nulle.

Zone d'intérêt non-pertinente (AOII) Une zone d'intérêt non-pertinente est une zone d'intérêt qu'il n'est pas nécessaire de regarder pour répondre à la question propre à l'image.

Zone d'intérêt nulle Une zone d'intérêt nulle est une zone d'intérêt qui peut être ignorée par le sujet (par exemple une zone se situant en dehors des données visuelles).

Zone d'intérêt pertinente (AORI) Une zone d'intérêt pertinente est une zone d'intérêt qu'il est primordial de regarder pour répondre à la question propre à l'image.

Table des matières

Table des matières	vii
Table des figures	xi
Liste des tableaux	xiii
1 Introduction	1
1.1 Problématique	2
1.2 Approche et méthodologie	3
1.3 Structure du document	5
I État de l’art	7
2 Les patrons de conception	9
2.1 Description générale	10
2.2 Propriétés des patrons de conception	11
2.3 Éléments constitutifs	12
2.4 Catégorisation	13
2.5 Qualité des patrons de conception	14
3 Compréhension et représentation	15
3.1 Compréhension des programmes	16
3.2 Représentation des design patterns	22
4 Science de la vision	29
4.1 Processus cognitif et théorie de la vision	30
4.2 Catégorisation des mouvements oculaires	30
4.3 Oculométrie	32
5 Mesure de l’effort de compréhension	41
5.1 La charge mentale de travail	42
5.2 Évaluation de la charge mentale de travail	42
5.3 NASA-TLX	43
6 L’expérimentation dans le génie logiciel	45
6.1 Génie logiciel empirique	46

6.2	Stratégies empiriques	47
6.3	Conception expérimentale	48
7	Matériel	61
7.1	Oculomètre	62
7.2	TAUPE	63
8	Distance relative entre chemins visuels	71
8.1	Principe	72
8.2	Distance de Levenshtein	72
8.3	Dynamic Time Warping	73
8.4	Étude de cas	77
II	Experimentation	81
9	Définition de l'expérience	83
9.1	Éléments constitutifs de l'expérience	84
9.2	Mise en place de l'expérience	99
10	Analyse de la tâche « composition »	105
10.1	Structuration de l'analyse des tâches	107
10.2	Sélection des métriques significatives	108
10.3	Analyse des mesures physiologiques	110
10.4	Analyse des distances relatives entre chemins visuels	115
10.5	Conclusion de l'analyse de la tâche	116
11	Analyse de la tâche « participation »	119
11.1	Sélection des métriques significatives	121
11.2	Analyse des performances	123
11.3	Analyse des mesures physiologiques	126
11.4	Analyse des distances relatives entre chemins visuels	137
11.5	Conclusion de l'analyse de la tâche	139
12	Analyse de la tâche « rôle »	141
12.1	Sélection des métriques significatives	142
12.2	Analyse des performances	144
12.3	Analyse des mesures physiologiques	145
12.4	Distances relatives entre chemins visuels	150
12.5	Conclusion de l'analyse de la tâche	151
13	Analyse des mesures subjectives	153
13.1	Approche et mise en place	154
13.2	Étude de l'influence des variables indépendantes	154
13.3	Étude de l'influence des facteurs confondants	156
13.4	Conclusions	158
14	Post expérience	159
14.1	Comparaison des résultats des tâches	160
14.2	Comparaison avec l'expérience de Porras	161

14.3	Évaluation de la représentation proposée	162
14.4	Menaces à la validité	164
III Conclusion		169
15	Conclusions et travaux futurs	171
15.1	Réponses aux questions d'exploration, de recherche et de prospection	171
15.2	Autres contributions	176
15.3	Travaux futurs	177
Bibliographie		179
A	Tableaux de données	183
A.1	Tableau de répartition sujet-représentation-framework	183
A.2	Tableau des mesures de performances	184
A.3	Tableaux des mesures physiologiques	189
A.4	Tableau des distances relatives entre chemins visuels	197
A.5	Tableau de la mesure subjective	205
B	Questionnaire de l'expérience	207
C	Questionnaire post-expérimental	227
D	Diagrammes de l'expérience	233
D.1	Nutch / <i>Dong</i>	233
D.2	Nutch / <i>SPE</i>	235
D.3	Nutch / <i>UMLNotation</i>	237
D.4	JRefactory / <i>Dong</i>	239
D.5	JRefactory / <i>SPE</i>	241
D.6	JRefactory / <i>UMLNotation</i>	243

Table des figures

3.1	Modèle de Brooks	20
3.2	Modèle de Soloway	21
3.3	Exemple d'une représentation UML Collaboration Notation	23
3.4	Exemple d'une représentation de Dong	24
3.5	Exemple d'une représentation de Schauer	25
3.6	Exemple d'une représentation de Gamma	26
3.7	Exemple d'une représentation Stereotype Pattern Enhanced	27
4.1	Observation d'une image suivant différentes questions posées [TWK+10].	31
4.2	Position de la réflexion cornéenne modifiée selon le point du regard [RL01].	33
4.3	Enveloppe convexe	35
4.4	Exemple de densité spatiale égale à 8% [DL11]	35
4.5	Exemple de parcourt visuel [DL11]	35
6.1	Le principe expérimental [WRH+00]	49
6.2	Illustration d'une expérience [WRH+00]	49
6.3	Activités du processus d'expérimentation [WRH+00]	50
6.4	Phase de planification d'une expérience [WRH+00]	52
6.5	Phase d'exécution d'une expérience [WRH+00]	55
6.6	Exemple de boîte à moustaches	57
7.1	L'oculomètre FaceLAB de l'École Polytechnique de Montréal	64
7.2	Architecture conceptuelle de TAUPE	66
7.3	Fenêtre principale du logiciel TAUPE	67
7.4	Zones d'intérêt et de non intérêt de la question « Q1 » et de la question « Q2 »	69
7.5	Heatmap basée sur la durée des fixations	70
8.1	Comparaison entre deux chemins visuels	74
8.2	Chemin de correspondances entre deux courbes dont les longueurs sont différentes	76
8.3	Chemin de correspondances où est appliquée la condition de monotonie. Comme on le voit, aucun retour en arrière n'est permis	77
8.4	Moyenne des distances relatives entre chemins visuels en fonction du niveau d'étude suivant les deux techniques (Levenshtein et DTW)	79
9.1	Une démonstration d'utilisation de Nutch	85
9.2	L'interface de JRefactory	86

9.3 Environnement de l'expérimentation.	94
10.1 Étude du taux IN AORI/IN AOII normalisé en fonction des représentations	110
10.2 Étude du ratio durée fixation/saccade en fonction de la représentation	112
10.3 Étude du pourcentage de transition AORI vers AOII en fonction de la représentation . . .	114
10.4 Étude de la distance relative entre chemins visuels en fonction du niveau d'étude des participants pour la tâche « composition »	116
11.1 Étude du passé sur la tâche « participation » en fonction de la représentation	123
11.2 Moyenne de l'exactitude des réponses de la tâche « participation » en fonction de la représentation	125
11.3 Étude de la densité spatiale en fonction de la représentation	126
11.4 Étude de la densité de transition en fonction de la représentation	128
11.5 Étude du taux IN AORI/IN AOII normalisé en fonction de la représentation - partie question	129
11.6 Étude du taux IN AORI/IN AOII normalisé en fonction de la représentation - partie support	132
11.7 Étude du ratio durée fixation/saccade en fonction de la représentation	133
11.8 Étude du pourcentage de transition AORI vers AOII en fonction de la représentation . . .	135
11.9 Étude du nombre moyen de fixations pos-AORI en fonction de la représentation	136
11.10 Étude de la distance relative entre chemins visuels en fonction du niveau d'étude des participants pour la tâche « participation »	138
12.1 Étude de l'exactitude des réponses de la tâche « rôle » en fonction de la représentation . .	144
12.2 Étude de la durée moyenne des fixations en fonction de la représentation	146
12.3 Étude du ratio durée fixation/saccade en fonction de la représentation	147
12.4 Étude du pourcentage de transition AORI vers AOII en fonction de la représentation . . .	149
12.5 Étude de la distance relative entre chemins visuels en fonction du sexe des participants pour la tâche « rôle »	151
13.1 Étude de la charge mentale subjective en fonction de la représentation	155
13.2 Étude de la charge mentale subjective en fonction du framework utilisé	156
13.3 Étude de la charge mentale subjective en fonction du sexe des sujets	157
13.4 Étude de la charge mentale subjective en fonction du niveau d'étude des sujets	158
D.1 Nutch / <i>Dong</i>	234
D.2 Nutch / <i>SPE</i>	236
D.3 Nutch / <i>UMLNotation</i>	238
D.4 JRefactory / <i>Dong</i>	240
D.5 JRefactory / <i>SPE</i>	242
D.6 JRefactory / <i>UMLNotation</i>	244

Liste des tableaux

3.1	Modèles cognitifs	17
4.1	Matrice de transition de la figure 4.5 [DL11]	35
6.1	comparaison des stratégies empiriques [WRH+00]	48
6.2	Résumé des tests d’hypothèses paramétriques/non-paramétriques pour les différents type de conception	59
7.1	Évolution des métriques de TAUPE en fonction des versions	64
9.1	Classification des patrons de conception selon Gamma <i>et al.</i> [GHJ+94]	87
9.2	Les six questions relatives au framework JRefactory	90
9.3	Les six questions relatives au framework Nutch	90
9.4	Attribution des zones d’intérêt de chaque question pour le framework JRefactory	97
9.5	Attribution des zones d’intérêt de chaque question pour le framework Nutch	98
9.6	Configuration des attributions des couples (<i>framework, représentation</i>)	101
10.1	Résumé des tests d’hypothèses des mesures de performance en fonction des variables indépendantes pour la tâche « composition »	108
10.2	Résumé des p-values des mesures physiologiques en fonction des variables indépendantes pour la tâche « composition »	109
10.3	Résumé des tests d’hypothèses sur la mesure de la distance relative entre les chemins visuels en fonction des facteurs confondants pour la tâche « composition »	110
10.4	P-value du taux IN AORI/IN AOII normalisé pour la partie « question » de la tâche « composition » en fonction des représentations	111
10.5	P-value du ratio durée fixation/saccade pour la partie « question » de la tâche « composition » en fonction de la représentation utilisée	113
10.6	P-value du pourcentage de transition AORI vers AOII pour la partie « question » de la tâche « composition » en fonction de la représentation utilisée	114
11.1	Résumé des tests d’hypothèses des mesures de performance en fonction des variables indépendantes pour la tâche « participation »	121
11.2	Résumé des p-values des mesures physiologiques en fonction des variables indépendantes pour la tâche « participation »	122
11.3	Résumé des tests d’hypothèses sur la mesure de la distance relative entre les chemins visuels en fonction des facteurs confondants pour la tâche « participation »	123

11.4	P-value du temps passé sur la tâche « participation » en fonction de la représentation utilisée	124
11.5	P-value du caractère exact des réponses de la tâche « participation » en fonction de la représentation utilisée	126
11.6	P-value de la densité spatiale pour la tâche « participation » en fonction de la représentation utilisée	127
11.7	P-value de la densité de transition pour la tâche « participation » en fonction de la représentation utilisée	128
11.8	P-value du taux IN AORI/IN AOII normalisé pour la tâche « participation » en fonction de la représentation utilisée - partie question	130
11.9	P-value du taux IN AORI/IN AOII normalisé pour la tâche « participation » en fonction de la représentation utilisée - partie support	133
11.10	P-value du ratio durée fixation/saccade pour la partie « question » de la tâche « participation » en fonction de la représentation utilisée	134
11.11	P-value du pourcentage de transition AORI vers AOII pour la partie « question » de la tâche « participation » en fonction de la représentation utilisée	136
11.12	P-value du nombre moyen de fixations post-AORI pour la partie « question » de la tâche « participation » en fonction de la représentation utilisée	137
12.1	Résumé des tests d'hypothèses des mesures de performance en fonction des variables indépendantes pour la tâche « rôle »	142
12.2	Résumé des p-values des mesures physiologiques en fonction des variables indépendantes pour la tâche « rôle »	142
12.3	Résumé des tests d'hypothèses sur la mesure de la distance relative entre les chemins visuels en fonction des facteurs confondants pour la tâche « rôle »	144
12.4	P-value du caractère exact des réponses de la tâche « rôle » en fonction de la représentation utilisée	145
12.5	P-value de la durée moyenne des fixations pour la partie « question » de la tâche « rôle » en fonction de la représentation utilisée	146
12.6	P-value du ratio durée fixation/saccade pour la partie « question » de la tâche « rôle » en fonction de la représentation utilisée	148
12.7	P-value du pourcentage de transition AORI vers AOII pour la partie « question » de la tâche « rôle » en fonction de la représentation utilisée	150
13.1	Poids attribué à chaque sous-échelle de NASA-TLX	154
A.1	Répartition sujet-représentation-framework	184
A.2	Données de performance	188
A.3	Données oculométriques - partie « question »	192
A.4	Données oculométriques - partie « support »	196
A.5	Données DRCV - sexe	201
A.6	Données DRCV - niveau d'étude	205
A.7	Données NASA-TLX	206

Introduction

Comme toute science, l'informatique, par sa complexité grandissante, s'est vue dans le besoin de se définir une base de connaissances et de techniques afin d'évoluer le mieux possible. Ces bases de connaissances peuvent concerner, par exemple, des sujets tels que la conception et l'évolution des logiciels. C'est dans cette approche qu'est né le génie logiciel, défini comme *l'ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi*¹. Le génie logiciel définit différents cycles de développement logiciel. Pour un logiciel d'une taille correcte, les différents cycles incluent *souvent*² les activités d'analyse des besoins, de conception, de développement, de test et de maintenance. Dans ce document, nous allons nous intéresser à l'activité de maintenance et plus particulièrement à la tâche de compréhension de programmes. Comme son nom l'indique, l'activité de maintenance se place à la fin du cycle de développement du logiciel. Cependant, cette phase se révèle être un aspect critique dans le sens où son coût peut devenir assez important.

Pour réaliser cette activité de maintenance, les informaticiens vont devoir comprendre le logiciel. En effet, une bonne **compréhension de programmes** permet de réaliser, de façon efficace, les tâches de développement et de maintenance. En réalité, ces tâches demandent une représentation mentale de l'architecture du programme. Pour faciliter la création de cette représentation mentale, des outils, tel les diagrammes, ont été proposés [KKB+98][Mal08].

Les diagrammes sont des outils visuels aidant à construire une représentation mentale du logiciel. En soulignant les informations pertinentes des objets et leurs relations, ils permettent d'avoir une vision globale et claire du logiciel [CK05].

Par ailleurs, les logiciels contiennent couramment des patrons de conceptions au sein de leur architecture. Cette présence est majoritairement due à la réutilisabilité qu'ils favorisent. Ceux-ci consistent à définir une solution globale en terme de schéma générique pour des problèmes se rapportant à un même contexte.

« Un design pattern fournit un schéma pour affiner les sous-systèmes, les composants d'un système software, ou les relations existantes entre eux. Il décrit une structure récurrente de composants communicants qui résout un problème général de design dans un contexte particulier ».

[GHJ+94]

1. Selon l'arrêté ministériel du 30 décembre 1983 relatif à l'enrichissement du vocabulaire de l'informatique [Journal officiel du 19 février 1984].

2. Rien ne permet d'affirmer que chaque cycle comprend ces activités.

Ces éléments d'architectures logicielles possèdent plusieurs propriétés qui favorisent une bonne compréhension logicielle : ils documentent des solutions existantes, offrent un vocabulaire commun et représentent un moyen de documentation logicielle. Par conséquent, ils constituent des éléments à prendre en compte lors de n'importe quelle tâche de compréhension logicielle. Or, ces patrons de conception ne sont pas toujours pris en considération par les techniques de représentations des artefacts logiciels. La représentation UML d'un diagramme de classes, par exemple, ne permet pas de visualiser les patrons de conception utilisés dans l'architecture. C'est de ce manque que vient la problématique de notre document.

1.1 Problématique

La compréhension de programmes est donc une étape préliminaire essentielle à toutes les activités de développement et de maintenance logiciels. Cette étape est difficile à cause de la complexité et de la taille des programmes, de la difficulté de leurs domaines d'applications et de la pression du temps. Des méthodes ont été proposées et de nombreuses approches ont été développées pour faciliter la compréhension de programmes.

Parmi ces méthodes, la rétro-conception des programmes inclut toutes les approches ayant pour objectif de fournir aux développeurs des représentations des artefacts logiciels (code, tests, documentation, etc.) de plus haut-niveau afin de leur permettre d'appréhender le logiciel sans se perdre dans les détails inutiles à leurs tâches.

Actuellement, la représentation standard de visualisation de patrons de conception UML collaboration notation. Plusieurs auteurs ont remarqué des lacunes dans cette représentation et ont proposé de nouvelles représentations pour remplacer le standard. En 2008, Cepeda Porras [Por08] a réalisé une étude empirique intitulée « *Analyse, à l'aide d'oculomètres, de techniques de visualisation UML de patrons de conception pour la compréhension de programmes* ». Cette analyse évalue l'efficacité des représentations visuelles de patrons de conception sur des diagrammes de classes UML. Elle a pour but de comparer la représentation standard avec trois représentations différentes. Les représentations utilisées sont : UML collaboration notation [Vli98], pattern-enhanced class diagrams (notée *Schauer*) [SK98], stereotype-enhanced UML diagrams (notée *Dong*) [DYZ07] et "pattern :role" notation (notée *Gamma*) [Gam96]. L'analyse des résultats démontre que la représentation stereotype-enhanced UML diagrams [DYZ07] semble être plus efficace que la représentation standard pour deux des trois tâches de compréhension de patrons de conception.

L'expérience de Cepeda Porras est donc une contribution concrète dans la compréhension de programmes. Malgré tout, aucune suite n'a été donnée aux conclusions qu'il a énoncées. Nous arrivons donc au cœur de notre sujet qui est l'« *Analyse de la compréhension des patrons de conception : expérimentation avec différentes représentations* ». Notre étude est en réalité une réponse à l'expérience menée par Cepeda Porras. En effet, cette dernière contient plusieurs limites et ne propose pas de nouvelle représentation. Pour pallier à ces limites, nous avons décidé de créer une nouvelle représentation, nommée *stereotype pattern enhanced UML diagrams (SPE)*. Cette dernière a été réalisée sur base des résultats d'études précédentes de Yusuf *et al.* [YKM07] et de Cepeda Porras [Por08]. Naturellement, nous avons réalisé une nouvelle expérience pour évaluer la représentation proposée et la comparer avec celles déjà prises en compte par Cepeda Porras.

Toute activité de recherche nécessite d'abord de se questionner sur l'état de l'art. Pour ce présent

document, ce questionnement nécessite de se pencher sur des sujets particuliers, tels que la théorie de la compréhension, la théorie des approches empiriques et les techniques de représentations de diagrammes de classes contenant des patrons de conception. Ce présent document a pour vocation de répondre à trois questions ayant un rapport avec ces théories :

Question 1 : Quelles théories et outils le génie logiciel offre-t-il pour mesurer la compréhension de la structure d'un programme contenant des patron de conceptions ?

Question 2 : Comment réaliser une approche empirique pour évaluer la compréhension ?

Question 3 : Quelles sont les alternatives et les variantes à la représentation des patrons de conception dans des diagrammes de classe UML ?

Une fois ces trois questions d'exploration traitées, nous pouvons nous intéresser aux questions de recherches proprement dites. Celles-ci traitent de l'analyse de l'existence d'un lien entre les représentations et l'efficacité de réalisation d'une tâche de compréhension de patrons de conception. Ainsi, nous dénombrons deux questions de recherches :

Question 4 : Est-ce qu'une nouvelle étude confirme l'influence des représentations détectée par Cepeda Porras ?

Question 5 : Pour quelles raisons la représentation Dong est-elle moins performante que la représentation UML Collaboration Notation lorsqu'il s'agit d'identifier les classes appartenant à un patron de conception alors qu'elle est plus performante pour détecter les patrons de conception dans lesquels une classe intervient ainsi que les différents rôles joués par une classe ?

Enfin, après avoir répondu à ces différents points, nous pouvons tenter de réfléchir aux améliorations qu'il est possible d'apporter à certaines représentations. Cette réflexion prospective nous amène à la question suivante :

Question 6 : Est-ce qu'une nouvelle représentation peut corriger les lacunes des autres ?

1.2 Approche et méthodologie

Pour réaliser cette « Analyse de la compréhension des patrons de conception : Expérimentation avec différentes représentations », nous devons suivre une approche empirique. De ce fait, nous avons donc réalisé une étude empirique pour collecter les données sur les performances de plusieurs sujets expérimentés en compréhension de programmes orienté-objet. Leur *mission* étant de répondre à des questions se référant à trois tâches de base dans la compréhension de patrons de conception³ tout en étant observés :

- La tâche de participation de classe qui se traduit par l'identification de toutes les classes qui participent à un patron de conception.
- La tâche de composition de patrons qui s'intéresse à l'identification de tous les patrons de conception auxquels une classe participe.
- La tâche d'identification des rôles joués par une classe qui consiste à identifier les différents rôles que joue une classe dans les patrons de conception.

3. Nous réalisons une étude afin de comprendre les conclusions de Porras, ainsi nous avons repris les trois mêmes tâches [Por08].

Comme son nom l'indique, le sujet de cette analyse repose essentiellement sur les représentations utilisées, nous les avons sélectionnées avec soin. Celles qui ont été retenues sont les suivantes : UML collaboration notation [Vli98], stereotype-enhanced UML diagrams [DYZ07] et stereotype pattern enhanced UML diagrams (SPE). Deux framework, JRefactory et Nutch, sont utilisés comme support pour l'expérience. Chaque sujet sera amené à répondre aux trois tâches citées ci-dessus, à partir de deux couples (framework, représentation). Chaque couple (framework, représentation) ayant le même framework proviendront d'un même diagramme de classes UML de base, avec la même disposition des classes et le même nombre de classes et de méthodes. Dans la mesure du possible, les diagrammes présentent les patrons de conception suivant la représentation canonique telle que décrite dans [GHJ+94]. Cependant, il existe de légères différences sur la mise en page, occasionnées principalement par des particularités de croisement d'arcs.

En ce qui concerne l'étude en soi, nous procédons méthodiquement pour analyser les résultats de notre expérience. Premièrement, nous effectuons une approche d'étude de la performance basée sur le pourcentage de réponses correctes et du temps employé par les sujets pour exécuter chaque tâche. Ensuite, une étude sur la compréhension des diagrammes par les sujets est réalisée à l'aide de la théorie Vision-Compréhension, telle que présentée par Guéhéneuc [Gué08]. Enfin, un système subjectif d'évaluation de la charge mentale permet au sujet d'exprimer sa perception quant à l'effort qu'il a fourni durant ses tâches.

Nous utilisons donc la théorie Vision-Compréhension pour réaliser une expérience sur la compréhension des différentes notations de représentations. En effet, afin de mesurer la charge cognitive de chaque sujet pendant l'exécution de l'expérience, nous utilisons des oculomètres. Pratiquement, un oculomètre enregistre les coordonnées du regard d'un sujet lorsqu'il regarde un écran. Il fournit une nouvelle perspective sur le processus de compréhension d'un sujet car il montre les zones qui attirent l'attention de celui-ci ainsi que le parcours visuel du regard sur l'écran [Duc07].

Si nous avons choisis cette théorie c'est pour plusieurs raisons. Premièrement, depuis quelques années, la théorie de la compréhension tente d'aider de telles expériences. Elle a été étudiée par de nombreux chercheurs, par exemple pour construire des modèles de compréhension de programmes, tels que [Bro78, vMV95, SFM99], ou des outils pour faciliter cette activité, par exemple [BG97]. Récemment, des chercheurs ont introduit l'utilisation d'appareils d'oculométrie pour améliorer la compréhension des processus cognitifs des développeurs qui prennent place lors de la compréhension des programmes [Gué06, Gué08, YKM07]. Cette utilisation de l'oculométrie a été, par exemple, appliquée à l'étude de données visualisées telles qu'un code source, des identificateurs [SM10], des patrons architecturaux [DL11], etc. Deuxièmement, l'étude dirigée par Cepeda Porras [Por08] a utilisé cette théorie et est arrivé à des résultats significatifs. Dernièrement, le milieu dans lequel nous avons évolué nous permettait aisément d'utiliser du matériel d'oculométrie.

Cependant, l'utilisation d'appareil d'oculométrie nous a obligés à nous intéresser davantage aux métriques existantes dans ce domaine. L'ouvrage réalisé par Poole et Ball [PB06] est une référence en terme de métriques oculométriques. Bien sûr, une sélection a du être effectuée. Une autre préoccupation inhérente à l'utilisation de ce matériel, provient des logiciels qui travaillent avec des fichiers générés par un oculomètre. Dans la plupart des cas, l'utilisation de logiciels spécifiques empêchent la liberté des chercheurs dans le sens que ces logiciels sont prioritaires et ne permettent donc pas d'être extensibles comme ils le souhaiteraient. Cette menace a été écartée, depuis quelques années,

par la réalisation du projet TAUPE⁴ programmé par et pour le laboratoire⁵ dans lequel nous étions affectés. Une contribution à ce logiciel, par nos soins, a été apportée.

1.3 Structure du document

Deux parties divisent ce document. La première partie concerne l'*État de l'art*. Cette partie présente les différents domaines et matériaux impliqués dans notre étude, et qui sont donc primordiaux à la compréhension de ce document.

Clé de voute de notre expérimentation, les patrons de conception sont définis dans le **chapitre 2**. Aucun patron n'est mis en valeur dans le sens qu'il n'y a pas de patron privilégié pour notre analyse. Ensuite, le **chapitre 3** introduit le concept de compréhension des programmes. De plus, les différentes représentations qui ont pour but de faciliter cette compréhension en mettant en valeur les patrons de conception sont également présentées. Cette présentation contient également la nouvelle représentation proposée. Ainsi, le lecteur comprend rapidement la portée de l'analyse.

Le **chapitre 4** s'intéresse plus particulièrement à la théorie de la vision. Le but est de familiariser le lecteur avec les nombreuses notions propres à ce domaine. De plus, les métriques relatives aux matériel d'oculométrie sont également présentées et expliquées.

Ensuite, une brève présentation du concept de la charge mentale ainsi qu'un outil pour la mesurer sont expliquées au **chapitre 5**. Cette technique permet de comprendre à quel point le développeur fait face à sa charge mentale, et donc sa difficulté de compréhension, pour mener à bien une tâche demandée.

Notre approche repose sur une étude empirique, dans le cadre du génie logiciel, dont la théorie est décrite au **chapitre 6**.

Elle se base sur du matériel et des logiciels relatifs au domaine de l'oculométrie. Ces différents éléments sont présentés et expliqués dans le **chapitre 7**.

Enfin, le **chapitre 8** introduit une amélioration apportée à une métrique oculométrique, se rapportant aux parcours visuels des participants. Cette nouvelle métrique est décrite formellement et ensuite appliquée à une étude de cas réelle. Concrètement, cette métrique est utilisée lors de l'analyse des résultats de notre expérience.

La deuxième partie est principalement consacrée à notre *étude empirique*. Avant de présenter les différentes analyses de cette étude, le **chapitre 9** définit précisément les différents éléments qui la constituent. Bien sûr, ce chapitre ne fait pas que de décrire mais prend la peine de justifier les différents choix effectués sur les éléments. Ce chapitre profite également de ce que les bases de l'expérience sont expliquées afin de présenter la mise en place exacte de celle-ci.

Une fois l'expérimentation définie, le lecteur trouvera les analyses des résultats de l'expérience, décomposées en quatre parties. Les trois premières parties se rapportent chacune à une tâche spécifique, tandis que la dernière partie est consacrée à l'analyse de l'évaluation subjective de la charge mentale.

Le **chapitre 10** s'intéresse à l'analyse de la tâche d'identification des patrons de conception propres à une certaine classe.

Ensuite, l'analyse de la tâche qui consiste à identifier toutes les classes participant à un patron de conception est présentée dans le **chapitre 11**.

Enfin, le **chapitre 12** contient l'analyse de la tâche d'identification de tous les rôles joués par une certaine classe.

4. <http://www.ptidej.net/research/taupe/>

5. <http://www.ptidej.net/>

Une dernière analyse est réalisée dans le **chapitre 13**. Celle-ci concerne l'étude de l'évaluation subjective de la charge mentale ressentie par les sujets lors des trois tâches.

Le **chapitre 14** fait une rétrospective de l'étude dans son intégralité, c'est-à-dire une comparaison des analyses des trois tâches, une comparaison des résultats obtenus avec ceux de Cepeda Porras [Por08], une critique de la nouvelle représentation ainsi que les menaces à la validité.

Enfin, le **chapitre 15** tire une conclusion de ce travail, met en avant notre contribution dans différents domaines, qu'il s'agisse de résultats d'expériences ou d'outils mis en place et propose différentes perspectives à explorer par la suite.

Première partie

État de l'art

Les patrons de conception

Les patrons de conception ont considérablement attiré l'attention des ingénieurs logiciel ces dernières années : des workshops, des forums, des tutoriel, des articles de journaux et des livres entiers sont consacrés à ce sujet. Ces patrons de conception sont au cœur de la problématique de ce document. La section 2.1 présente au lecteur une description générale de ceux-ci. Cette description aborde des sujets tels que l'historique et les définitions du terme patron. Les sections 2.2 et 2.3 reprennent respectivement les propriétés et les éléments constitutifs des patrons de conception. Deux techniques de classification des patrons sont proposées en section 2.4. Pour finir, les qualités qui peuvent être attendue de ceux-ci sont décrites en section 2.5.

Les différentes sections de ce chapitre sont majoritairement inspirées des ouvrages de Gamma et al. [GHJ+94] et de Buschmann et al. [BMR+96].

Sommaire

2.1	Description générale	10
2.1.1	Historique	10
2.1.2	Définition de patron	10
2.1.3	Patrons architecturaux	11
2.1.4	Patrons de conception	11
2.2	Propriétés des patrons de conception	11
2.3	Éléments constitutifs	12
2.4	Catégorisation	13
2.5	Qualité des patrons de conception	14

2.1 Description générale

Lorsque des experts travaillent sur des problèmes particuliers, il est souvent inhabituel pour eux d’y faire face en inventant une nouvelle solution qui est complètement distincte d’une autre pré-existante. Ce nouveau problème évoque souvent une difficulté similaire qu’ils ont déjà résolue. Ils peuvent donc réutiliser l’essence de la solution précédente pour résoudre le problème courant.

Les patrons de conception¹ aident les concepteurs à construire des programmes en se basant sur l’expérience d’ingénieurs logiciel qualifiés. Cette représentation des techniques éprouvées sous forme de design patterns les rend plus accessibles aux développeurs de nouveaux systèmes. Ces *patrons* favorisent le choix d’une alternative de conception qui rend le système réutilisable et évitent d’autres alternatives qui pourraient compromettre la réutilisation.

2.1.1 Historique

L’apparition du mot patron ou *motif*, traduction française du terme *pattern*, remonte aux années septante suite aux travaux de l’architecte Christopher Alexander [AIS+77]. A l’origine, le concept se présente comme étant propre à l’architecture. En 1987, Kent Beck et Ward Cunningham commencent à expérimenter l’utilisation de *pattern* dans la programmation. Dans le courant de la même année, ceux-ci présentent leurs résultats lors de la conférence OOSPLA [KC87]. Les patrons de conception gagnent finalement leur popularité au sein de l’informatique en 1994, lorsque Gamma *et al.* rédige le livre « *Design Patterns : Elements of Reusable Object-Oriented Software* » [GHJ+94]. Celui-ci prend la forme d’un catalogue de 23 patrons de conception. Deux années plus tard, Buschmann *et al.* [BMR+96] présentent un ouvrage dans lequel ils exposent la distinction de 3 terminologies de *pattern* au sein du génie logiciel : les *patrons architecturaux*, les patrons de conception et les *idioms*.

2.1.2 Définition de patron

Dans son ouvrage, Alexander nous dévoile une définition générale du terme « *pattern* » :

« Chaque patron décrit un problème qui se produit maintes et maintes fois dans notre environnement, et expose ensuite le coeur de la solution à ce problème, de telle sorte que vous pouvez utiliser cette solution un million de fois, sans jamais le faire de la même façon à deux reprises. » [AIS+77]

Buschmann *et al.* nous apportent quant à eux une définition propre aux architectures logicielles, mais qui comporte les mêmes idées :

« Un patron pour l’architecture logicielle décrit un problème de conception récurrent et particulier qui se pose dans des contextes spécifiques de conception, et qui présente un schéma générique éprouvé pour sa solution. Le schéma de solution est spécifié en décrivant ses éléments constitutifs, leurs responsabilités et les relations, et les moyens par lesquels ils collaborent. » [BMR+96]

D’après les deux définitions citées ci-dessus, nous pouvons relever le fait que chaque patron est une abstraction en trois parties qui révèle une relation entre :

- un contexte,
- un problème,
- une solution.

1. Dans la suite de ce document nous utiliserons de manière similaire les termes *design pattern* et patron de conception.

2.1.3 Patrons architecturaux

Des architectures logicielles viables sont construites selon certains principes de constructions généraux. Ceux-ci sont décrits via des patrons architecturaux. Buschmann *et al.* définissent un patron architectural de la manière suivante :

« Un patron architectural exprime un schéma fondamental d'organisation structurelle pour les systèmes logiciels. Il fournit un ensemble de sous-systèmes prédéfinis, précise leurs responsabilités et comprend des règles et des directives pour l'organisation des relations existantes entre eux. » [BMR+96]

Les patrons architecturaux sont des modèles permettant de construire des architectures logicielles précises. Ils spécifient les propriétés structurelles à l'échelle système d'une application et ont un impact sur l'architecture des ses sous-systèmes. Le pattern « Modèle-Vue-Contrôleur » est l'exemple le plus connu des patrons architecturaux.

2.1.4 Patrons de conception

Les sous-systèmes des architectures logicielles, ainsi que les relations entre eux, sont généralement constitués de quelques composants architecturaux plus petits. Les patrons de conception sont utilisés pour construire ces composants.

Une définition complète de ce type de patron est présente dans le livre de Gamma *et al.* :

« Un design pattern fournit un schéma pour affiner les sous-systèmes, les composants d'un système software, ou les relations existantes entre eux. Il décrit une structure récurrente de composants communicants qui résout un problème général de design dans un contexte particulier. » [GHJ+94]

Les patrons de conception sont donc des modèles à échelle moyenne. Ils sont plus petits en dimension que les patrons architecturaux, mais tentent d'être indépendants de tout langage de programmation ou paradigme de programmation. L'application d'un design pattern n'aura ainsi aucune influence sur la structure fondamentale du logiciel. Il aura, en contrepartie, un effet sur l'architecture des sous-systèmes. Nous pouvons citer comme exemple les patrons : « Observer », « Singleton », « Prototype », etc.

Les sections qui suivront ne traiteront uniquement que des patrons de conception, les patrons architecturaux n'ayant été définis qu'à titre informatif.

2.2 Propriétés des patrons de conception

La façon dont les patrons de conception répondent aux objectifs du génie logiciel est un critère important pour leur succès. Afin de ne pas être qu'un concept intellectuel intéressant, les patrons doivent aider au développement, à la maintenance et à l'évolution des systèmes à grande échelle. L'ensemble des propriétés des patrons de conception énoncées ci-dessous permet de répondre à ces objectifs.

Propriété 1 : les patrons de conception documentent des connaissances existantes et nous aident à trouver des solutions appropriées aux problèmes de conception. Ceux-ci sont issus de réutilisations d'architectures ou de conceptions qui ont été marquées par le succès.

Propriété 2 : les patrons fournissent une approche intéressante pour développer des logiciels possédant des propriétés définies. Ils permettent d'aider à implémenter les fonctionnalités d'une application en fournissant un squelette de comportement fonctionnel.

Propriété 3 : les patrons fournissent un niveau d'abstraction au dessus de celui des classes et des instances ou des composants. Tous les éléments décrits au sein du patron résolvent le problème auquel il s'adresse. Cette résolution se fait grâce aux différentes responsabilités des éléments et aux relations définies entre eux.

Propriété 4 : les patrons peuvent être entrelacés les uns aux autres. Nous pouvons les utiliser pour affiner d'autres patterns plus larges. Mais nous pouvons aussi les combiner pour résoudre des problèmes plus complexes.

Propriété 5 : les patrons peuvent être implémentés avec presque tous les langages et les paradigmes de programmation. Il est vrai que beaucoup de patrons utilisent des techniques propres à certains langages de programmation comme le polymorphisme ou l'héritage. Mais au niveau de la conception, les design patterns nécessitent seulement certaines facilités d'abstraction du langage, telles que les modules ou l'abstraction de données.

Propriété 6 : les patrons représentent un moyen de documentation des applications. Ils fournissent une spécification des différents modules, des interactions qui existent entre eux et des buts sous-jacents.

Propriété 7 : les patrons offrent un vocabulaire commun. Le nom d'un patron de conception se réfère à un problème, une solution et à ses conséquences. Il nous permet de concevoir à un niveau d'abstraction plus haut.

Propriété 8 : les patrons contribuent à la conception d'architectures logicielles hétérogènes. Chaque pattern apporte une solution à un problème dans un certain contexte à un bas niveau. L'entrelacement de plusieurs design patterns permet donc de concevoir des architectures hétérogènes.

Propriété 9 : les patrons aident à concevoir une application favorable aux changements. Chaque patron de conception laisse certains aspects de la structure du système varier indépendamment d'autres aspects. De cette manière, ils nous aident à ajouter de la souplesse dans nos programmes.

2.3 Éléments constitutifs

Comme nous avons pu le voir au sein du point 2.1.4, le patron met en valeur un contexte et un problème afin de proposer une solution. Ces trois termes sont essentiels et font partie des éléments constitutifs des patrons de conception.

Ces éléments, Gamma *et al.* [GHJ+94] en distinguent quatre : le nom, le problème, la solution et les conséquences. Buschmann *et al.* [BMR+96] quant à eux énoncent le contexte, le problème et la solution comme éléments constitutifs. La différence entre ces deux visions n'est pas réellement dérangeante, Gamma *et al.* rassemblent en fait le contexte et le problème en un seul élément.

Sur base de ces deux avis, nous pouvons alors relever la présence de cinq parties :

1. le nom du pattern est un moyen utilisé pour décrire un problème, une solution et ses conséquences en un mot ou deux. Nommer un pattern augmente directement notre vocabulaire de conception. Gamma *et al.* [GHJ+94] précisent que la recherche des noms des différents patrons de conception se trouvant dans leur catalogue a été l'étape la plus difficile.
2. le contexte décrit la situation dans laquelle le problème apparaît. Le contexte peut être général. Il peut également lier des patterns spécifiques [BMR+96].
3. le problème décrit quand un patron de conception peut être utilisé. Il prend la forme d'une difficulté répétitive dans un contexte donné, par exemple : comment représenter des algorithmes comme des objets ? Parfois, le problème inclut certaines conditions qui doivent être remplies pour pouvoir appliquer le patron [BMR+96].

4. la solution montre comment résoudre le problème. Elle décrit les éléments qui composent la conception, leurs relations, leurs responsabilités et leurs collaborations. Chaque pattern spécifie la structure, c'est-à-dire la configuration spatiale de ces éléments. La solution détermine ensuite un comportement à l'exécution [BMR+96].
5. les conséquences sont les résultats et les compromis résultants de l'utilisation du pattern. La réutilisation étant un facteur important dans la conception des applications en orienté objet, les conséquences d'un patron incluent son impact sur la flexibilité, la portabilité et l'extensibilité du système [GHJ+94].

Pour conclure, Buschmann *et al.* [BMR+96] précisent que l'intégralité des patrons fournissent une « boîte à outils mentale » qui nous aide à construire un logiciel respectant les exigences fonctionnelles et non-fonctionnelles. Gamma *et al.* relèvent également le fait que l'interprétation de ce qui est ou n'est pas un pattern est affecté par chaque point de vue. Au sein de leur livre [GHJ+94], ils ne se sont concentrés que sur les patterns avec un certain niveau d'abstraction.

2.4 Catégorisation

Les design patterns varient dans leur granularité et leur niveau d'abstraction. Comme il existe un grand nombre de patrons de conception, il est important de les classer. Pour se faire, deux approches ont été envisagées :

1. Gamma *et al.* [GHJ+94] classifient selon le **but** (créationnel, comportemental ou structurel) et la **portée** (classe ou objet) des design patterns.
2. Buschmann *et al.* [BMR+96] classifient les design patterns selon les **propriétés similaires** qu'ils aident à maintenir.

Concentrons-nous d'abord sur la première approche. Le critère de but comprend trois sous-catégories :

- créationnel : concerne les processus de créations d'objets.
- comportemental : caractérise la méthode avec laquelle les classes ou les objets vont interagir et distribuer les responsabilités.
- structurel : concerne la composition de classes ou d'objets.

Les patrons de conception sont ensuite classifiés suivant la portée :

- les patrons de classe agissent sur les relations entre les classes et les sous-classes. Ces relations sont établies à partir du principe d'héritage. Elles sont donc statiques et fixées lors de la compilation.
- les patrons d'objet se basent sur les relations entre les objets. Ces dernières sont donc dynamiques et peuvent être changées en cours d'exécution.

Comme nous l'avons énoncé plus haut, cette technique n'est pas la seule. Buschmann *et al.* [BMR+96] regroupent les patrons de conception en cinq familles différentes :

- décomposition structurelle : cette catégorie inclut les patterns qui soutiennent une décomposition appropriée des sous-systèmes et des composants complexes en parties coopérantes.
- organisation du travail : cette catégorie comprend les patrons qui définissent la façon dont les composants vont collaborer ensemble pour résoudre un problème complexe.
- contrôle d'accès : ces patterns gardent et contrôlent l'accès à des services ou des composants.

- gestion : cette catégorie comprend les patterns manipulant des collections homogènes d'objets, de services et de composants dans leur intégralité.
- communication : les patrons de cette catégorie aident à organiser les communications entre les composants.

2.5 Qualité des patrons de conception

Après avoir présenté une description générale des designs patterns, il est nécessaire de s'intéresser aux apports réels de valeur ajoutée de ceux-ci. Quels attributs de *qualité logiciel* les patrons peuvent-ils apporter ? Au sein de leurs ouvrages, Gamma *et al.* [GHJ+94] et Buschmann *et al.* [BMR+96] ne se réfèrent à aucun modèle de qualité logiciel. Malgré tout, ils mentionnent souvent des attributs tels que la *flexibilité*, la *compréhensibilité* ou la *maintenabilité*, sans donner de définition claire à ces concepts [Wen01].

De nombreux faits prouvent que les patterns ont un impact bénéfique sur la qualité des logiciels, mais dans certains cas ils peuvent aussi être utilisés de manière inappropriée en raison d'un manque d'expérience [Wen01]. Certaines études empiriques ont été conçues afin de relever des faits et des théories concernant les bénéfices en termes de qualité logiciel des design patterns.

Prenons comme exemple une première étude mise en place par Prechelt *et al.* [PUT+01]. Ceux-ci cherchent une preuve empirique exprimant les bénéfices de l'utilisation des patrons de conception. Leur expérimentation examine des scénarios de maintenance qui emploient divers design patterns et les compare à des alternatives de design plus simples. Les résultats révèlent des bénéfices lors de l'utilisation de design patterns, moyennant une expérience et une compréhension approfondie dans les patrons utilisés. Les auteurs soulignent néanmoins qu'il arrive que certaines situations soient plus propices à l'utilisation d'une solution plus simple.

Au sein d'une autre expérience, Prechelt *et al.* [PUP+02], investissent la nécessité ou non de documenter explicitement les patrons de conception au sein du code source des programmes. L'expérience compare la réalisation de tâches de maintenance sur deux programmes utilisant des design patterns :

- un comportant une documentation explicite des patterns utilisés,
- un sans documentation explicite.

Suivant leurs résultats, les auteurs concluent que la documentation explicite des patrons de conception réduit le temps nécessaire pour modifier un programme et aide à améliorer la qualité des modifications. Néanmoins, ils précisent que les bénéfices dépendent du programme, de la tâche de maintenance et de la connaissance du personnel en design pattern.

Différentes études empiriques ou encore différentes observations d'utilisation de patron de conception exposent leurs avantages. Certains de ceux-ci peuvent être énoncés à titre d'exemple : l'encouragement à la réutilisation de bonnes pratiques, la représentation d'un bon moyen de communication, ou encore la maintenabilité accrue. Ces expériences prouvent que ces *modèles de conceptions* ne sont pas seulement un concept intellectuel intéressant. Cependant, il est important de garder à l'esprit l'existence d'inconvénients tels que : la complexité d'utilisation, la consommation de temps et l'apprentissage difficile.

Compréhension et représentation

Le pilier central de ce document repose sur la « compréhension des programmes ». La compréhension de programmes est un sujet très présent dans le domaine du génie logiciel. Les nombreuses recherches qui en découlent ont dégagé des définitions sur le sujet et sur certains concepts inhérents. Des théories sur le processus de compréhension de programme ont même fini par voir le jour. L'ensemble de ces différents éléments sont décrits en section 3.1.

Une fois le domaine de la compréhension assimilé, il est important d'introduire des mécanismes créés par l'informaticien afin de faciliter cette compréhension. Nous allons donc nous intéresser aux design patterns, ou plus précisément à la représentation des design patterns. Si les design patterns sont reconnus pour être une bonne pratique de codage, qu'en est-il de l'aide qu'ils apportent à la compréhension des programmes ? Pour aider les développeurs, plusieurs présentations ont été créées. Ces représentations seront présentées en section 3.2.

La section 3.1 est majoritairement inspirée de Guéhéneuc Y. [Gué08], Rugaber S. [Rug95] et de Mayrhauser A. [Mar95] tandis que la section 3.2 se base des travaux de Dong et al. [DYZ07] et de Cepeda Porras G. [Por08]

Sommaire

3.1	Compréhension des programmes	16
3.1.1	Concept et terminologie	16
3.1.2	Les modèles	19
3.2	Représentation des design patterns	22
3.2.1	UML Collaboration Notation	22
3.2.2	Stereotype-enhanced UML diagrams	23
3.2.3	Pattern-enhanced class diagrams	25
3.2.4	« Pattern :role » notation	26
3.2.5	Stereotype Pattern Enhanced	26

3.1 Compréhension des programmes

La **compréhension des programmes** est un domaine du génie logiciel qui cherche à expliquer comment les informaticiens comprennent les programmes qu'ils doivent modifier (maintenir et/ou faire évoluer), c'est-à-dire comment ils obtiennent une représentation mentale de la structure ainsi que de la fonction d'un programme. Les implications de la compréhension d'un programme se situent au cœur de toutes les activités liées à ce logiciel, du développement à la maintenance, en passant par le déploiement et l'utilisation.

Il n'y a pas une définition unique sur la compréhension des programmes. Cependant, toutes les définitions s'accordent pour placer ce processus de compréhension comme étant un processus vital au bon développement d'un logiciel. Ci dessous, une définition de cette compréhension donnée par Spencer Rugaber :

« *La compréhension des programmes* est le processus d'acquisition de connaissances sur un programme informatique. L'amélioration des connaissances permet des activités telles que la correction de bug, l'amélioration, la réutilisation et la documentation. Bien que des efforts sont en cours pour automatiser le processus de compréhension, ces quantités importantes de connaissances et d'analyses qui sont nécessaires afin de faire ce processus qu'est la compréhension du programme est, aujourd'hui encore, en grande partie une tâche manuelle. »

[Rug95]

La compréhension des programmes est donc un facteur très important pour parvenir à une maintenance efficace des programmes et établir une évolution réussie de ces logiciels. Depuis plusieurs années, les chercheurs ont essayé de comprendre comment les programmeurs appréhendent les programmes durant la maintenance et l'évolution de ceux-ci. De nombreuses études sur la compréhension de programmes ont été publiées dans la littérature.

Cependant, la compréhension des programmes nécessite plus que la connaissance des faits, elle nécessite une théorie. Cinq types de tâches sont communément associées à la maintenance et à l'évolution des programmes : la maintenance adaptative, la maintenance perfective, la maintenance corrective, la réutilisation, et l'apprentissage du code. Certaines activités, comme *la compréhension du système ou du problème*, sont communes à plusieurs tâches. Pour analyser le processus cognitif qui se trouve derrière ces cinq tâches, certains auteurs ont développé plusieurs modèles, présentés dans le tableau 3.1 .

La compréhension des programmes est un domaine très vaste. En effet, étant donné son implication dans toutes les activités du logiciel, ce processus ne peut pas être étudié d'une façon isolée. Cette complexité implique la difficulté qu'ont les chercheurs à trouver une solution générale capable de répondre efficacement à ce besoin de compréhension.

3.1.1 Concept et terminologie

Comme énoncé précédemment, la compréhension des programmes implique certains pré-requis, cette partie s'adresse plus particulièrement à ces différents éléments qu'utilisent les programmeurs pour arriver à ce processus de compréhension.

Les connaissances

Les programmeurs possèdent deux types de connaissances : les connaissances générales qui sont indépendantes du logiciel qu'ils tentent de comprendre et les connaissances spécifiques qui représentent leur niveau de compréhension du logiciel.

Modèle	Activité de maintenance	Auteurs
Control-flow	Compréhension	Pennington
Functional	Compréhension	Pennington
Top-down	Compréhension	Soloway, Adelson et Ehrlich
Integrated	Compréhension, maintenance corrective, adaptative et perfective	Von Mayrhauser and Vans
Autres	Amélioration	Letovsky
	Maintenance corrective	Vessey
	Compréhension	Brooks
		Shneiderman and Mayer

Tableau 3.1 – Modèles cognitifs

Durant ce processus de compréhension, les programmeurs vont acquérir plus de connaissances spécifiques du logiciel mais peuvent aussi avoir besoin davantage de connaissances générales.

Les connaissances existantes concernent les langages de programmations, l'environnement, les principes de programmation, le domaine spécifique des choix d'architecture, les algorithmes et les différentes approches des solutions possibles.

Les nouvelles connaissances concernent principalement le produit logiciel. Elles sont acquises tout au long du processus de compréhension du code. Ces connaissances sont liées à la fonctionnalité, à l'architecture logicielle, à la façon dont les algorithmes et les objets sont mis en œuvre, au contrôle, au flux des données, etc. Cela s'étend sur de nombreux niveaux d'abstraction (de la constatation qu'« il s'agit d'un système d'exploitation » à « la variable *i* est incrémenté dans cette boucle »).

Le processus de compréhension consiste à faire correspondre ces connaissances du logiciel à leurs connaissances existantes jusqu'à ce que les programmeurs croient comprendre le code. Le modèle mental représente l'ensemble de ces correspondances. Il peut être complet ou incomplet..

Le modèle mental

Le modèle mental est une représentation interne du logiciel en cours d'examen. Il contient des entités statiques telles que des structures de texte, des blocs, des hypothèses de plans, des balises et des règles de discours. Les *plans à haut niveau*¹ affinent plus ces entités en les recomposant en blocs. Chacun de ces blocs représente une abstraction de niveau plus élevé d'autres morceaux ou des structures de texte. Une construction de blocs combine ainsi plusieurs comportements dynamiques, y compris des stratégies, des actions, des épisodes, et des processus [Mar95].

Les caractéristiques des experts

Un programmeur avec un certain niveau d'expertise dans un domaine donné affecte grandement la compréhension du programme. Les experts ont tendance à développer les caractéristiques suivantes :

- Ils organisent des structures de connaissances par rapport aux caractéristiques fonctionnelles du domaine dans lequel ils sont experts. Par exemple, à l'inverse des novices qui vont comprendre un programme en fonction de la syntaxe du programme, les experts pourraient organiser les connaissances du programme sous forme d'algorithmes.

1. Top-level en anglais.

- Les experts ont développé des schémas spécialisés qui font souvent abstraction des systèmes conçus antérieurement.
- Ces schémas spécialisés contribuent à la décomposition efficace et à la compréhension des problèmes. La compréhension « top-down » devient possible pour des problèmes qui correspondent à ces schémas spécialisés.
- Les experts ont une approche plus flexible vis-à-vis de la compréhension des problèmes. Ils écartent les hypothèses douteuses beaucoup plus rapidement que les novices, et ils ont tendance à générer en premier lieu une vue en largeur du programme. Quand de plus amples informations sont disponibles, ils affinent leurs hypothèses.

Psychologie du logiciel

L'étude de la psychologie du logiciel a été mise au point par **Shneiderman**. Il tente de découvrir et de décrire les limites de l'homme en interaction avec les ordinateurs. Shneiderman définit la psychologie du logiciel en tant qu' « étude de la performance humaine dans l'utilisation de l'ordinateur et des systèmes d'information ». Il utilise les techniques de la psychologie expérimentale afin d'analyser les aspects de la performance humaine dans des tâches informatiques.

Il applique également les concepts de la psychologie cognitive aux processus cognitifs et perceptifs impliqués dans l'interaction avec les ordinateurs. Dans le cas de la maintenance, la compréhension des compétences humaines et la capacité de travailler avec un logiciel sont nécessaires afin de faciliter l'examen du programmeur et la compréhension du code source. Les points forts et les limites des capacités de l'homme servent donc comme facteurs dans la détermination de la fonctionnalité des outils de maintenance de logiciels.

La psychologie du logiciel se concentre sur les facteurs humains tels que : la facilité d'utilisation, la simplicité de l'apprentissage, l'amélioration de la fiabilité, la réduction de la fréquence d'erreur, et l'amélioration de la satisfaction des utilisateurs. La compréhension des programmes est étudiée via des expériences demandant la mémorisation et la reconstruction de programmes. Une tâche de mémorisation/reconstruction consiste à étudier un programme pour ensuite le reconstruire à partir de la mémoire.

Comme expliqué dans le paragraphe sur l'expertise, il a été constaté, durant cette expérimentation de Shneiderman, que l'expérience des sujets joue un *rôle essentiel* dans cette tâche. Les programmeurs expérimentés écrivent des versions qui fonctionnellement sont équivalentes, mais qui sont syntaxiquement différentes. L'auteur explique cela par le fait que les sujets possédant de l'expérience dans la programmation améliorent leur capacité à reconnaître des structures significatives de programmes, ce qui leur permet de recoder la syntaxe dans une structure sémantique de niveau supérieur. Après que le sujet ait effectué la tâche de reconstruction, l'expérimentateur a analysé le code reconstruit en termes de blocs d'information. Ces blocs donnent un aperçu des types de structures internes d'information contenues dans le code. C'est en fonction de ces structures internes que le code est interprété ou compris par les sujets. Shneiderman identifie ces structures dans son modèle.

3.1.2 Les modèles

Basé sur l'étude de la psychologie du logiciel, une variété de modèles a été proposée pour la compréhension des programmes (voir tableau 3.1). Deux sont décrits ici pour illustrer les enjeux de la modélisation de ce comportement complexe. Ils ont été développés par **Ruven Brooks** et **Elliot Soloway**. La structure de base de ces deux modèles se compose de quatre composants :

1. Le système cible, c'est-à-dire le système que l'on veut comprendre. Il représente toutes les sources d'information disponibles pour la personne chargée de comprendre ce système, comme le code source et la documentation.
2. La base de connaissances qui accompagne l'expérience du sujet et ses connaissances de base utilisées dans la tâche de compréhension. Cette connaissance de base est soit interne (on parle alors de l'esprit du sujet), soit externe (par exemple dans le manuel de référence d'un langage de programmation).
3. Le modèle mental qui enregistre l'état actuel des connaissances du programme cible. Il est constamment mis à jour au cours de la compréhension.
4. Un processus d'assimilation qui interagit avec les trois autres composantes pour mettre à jour l'état actuel de la compréhension.

Les différences entre les deux modèles de la compréhension humaine sont dans la terminologie décrivant le contenu de la base de connaissances et dans l'approche que chacun adopte pour le processus d'assimilation. Alors que certains modèles contiennent des mécanismes qui utilisent les approches « **top-down** » et « **bottom-up** » pour aborder le processus de la compréhension, le modèle de Brooks souligne l'approche « top-down », alors que le modèle Soloway est plus « bottom-up ».

Le modèle de Brooks

Le modèle de Ruven Brooks traite de la compréhension des programmes réalisés. Il trouve son fondement dans des zones en dehors de l'informatique, tels que la résolution d'un problème thermodynamique, la résolution d'un problème de physique ou les échecs. Le modèle a été initialement créé pour expliquer les **quatre principales sources** de variation observées dans le processus de compréhension des programmes :

1. *Source* : La fonctionnalité du programme doit être comprise. Comment les différents programmes (qui dépendent de leur calcul) varient dans cette compréhensibilité ?
Explication : La complexité intrinsèque du domaine de travail (par exemple la physique nucléaire par rapport à la facturation) provoque des hypothèses de niveau supérieur qui sont plus complexes. Cela se traduit par une utilisation d'un plus grand nombre de domaines intermédiaires entre le domaine de travail et le domaine de programmation. En outre, la documentation expliquant ces domaines intermédiaires est plus rare que la documentation portant sur la tâche du programme original.
2. *Source* : Les différences par rapport aux textes des programmes. Pourquoi les programmes qui sont écrits dans différents langages diffèrent dans la compréhensibilité, et ce, même si c'est le même calcul qui est effectué.
Explication : Le code et la documentation affectent la facilité de trouver les balises et la facilité

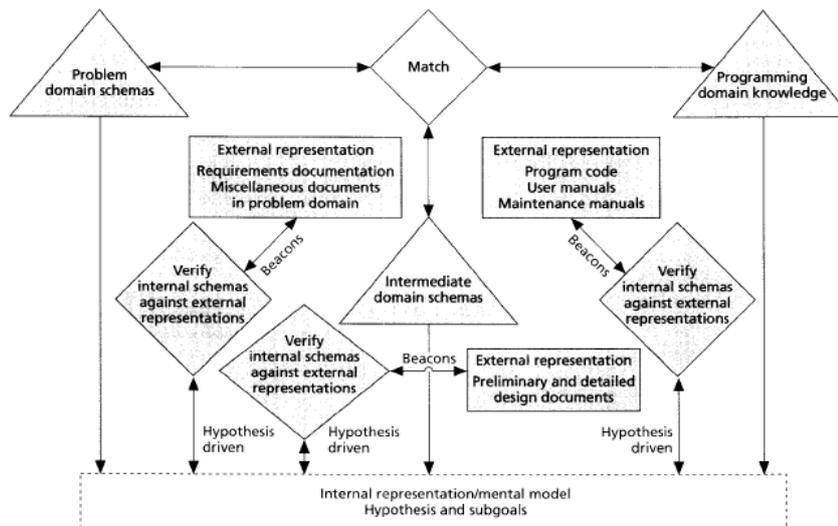


FIGURE 3.1 – Modèle de Brooks

de lier le code source à des hypothèses. En outre, les caractéristiques du langage influencent la confirmation des balises. Par exemple, l'hypothèse selon laquelle la variable « PI » contient « 3,1415 » est facile à confirmer dans le langage C si « PI » est définie en utilisant une déclaration *const*. Cela n'est pas si facile dans un langage qui ne possède pas des déclarations de constantes. En effet, même si au début du programme, « 3,1415 » est placé dans « PI », il faut encore prouver que « PI » n'est jamais changé.

3. *Source* : La motivation du programmeur qui doit comprendre le programme. Pourquoi le processus de compréhension varie selon que sa motivation soit pour déboguer le programme ou pour l'améliorer ?

Explication : Cette motivation affecte la stratégie utilisée dans la création et dans le suivi des hypothèses. Par exemple, le débogage d'une erreur de format pour un « output » provoque des hypothèses secondaires, telles que celles traitant de l'« input » ou du calcul, restées inexplorées en raison du manque de pertinence vis-à-vis de la tâche à accomplir.

4. *Source* : Les différences individuelles entre les capacités des programmeurs à comprendre le but d'un programme. Pourquoi un programmeur trouve un programme plus facile à comprendre par rapport à un autre programmeur ?

Explication : La connaissance du programmeur sur les domaines affecte le processus à tous les niveaux. Plus précisément, la connaissance du domaine de la tâche influe sur la qualité des hypothèses primaires/de haut-niveau. Tandis que la connaissance du domaine de programmation affecte les liaisons de niveau inférieur et les processus de localisation des balises.

Le modèle de Soloway

Le modèle d'Elliot Soloway traite également de la compréhension des programmes, mais divise la connaissance de base et le processus d'assimilation différemment. Dans la terminologie de Soloway, comprendre un programme est synonyme de récupérer l'intention qui est derrière le code.

Les objectifs désignent des intentions, et *les plans* désignent les techniques pour la réalisation de ces intentions. Les plans de travail sont comme des règles de réécriture qui transforment les objectifs

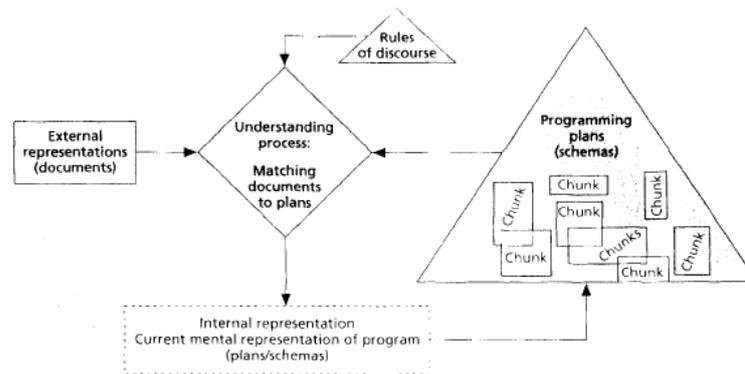


FIGURE 3.2 – Modèle de Soloway

en sous-objectifs, et enfin en code. La compréhension des programmes est définie comme étant le processus de reconnaissance des plans du code. En combinant ces plans et en inversant les règles de réécriture, on forme les sous-objectifs. Ensuite, on combine les sous-objectifs en objectifs de niveau supérieur.

La base de connaissances utilisée dans ce modèle contient :

- La sémantique des langages de programmation. Cela traite de la connaissance du programmeur du langage dans lequel le programme cible a été écrit.
- Objectif des connaissances. C'est l'encodage de l'ensemble des objectifs de calcul du programmeur. Les objectifs sont encodés indépendamment des algorithmes et des langages qui les implémentent.
- Le plan des connaissances. C'est l'encodage des solutions aux problèmes que le programmeur a résolus ou compris dans le passé. Ces solutions sont des composantes de bas niveau et comprennent celles qui sont universellement connues des programmeurs. Le plan de connaissances prévoit également que le programmeur a acquis des connaissances spécifiques au domaine.
- Efficacité des connaissances. Cela montre comment les programmeurs détectent les inefficacités ainsi que l'influence qu'ont les questions d'efficacité sur le code de programmation et les plans.
- La connaissance du domaine du problème. C'est la connaissance qu'a le programmeur sur le domaine de l'application jusqu'au domaine de calcul ainsi que ces domaines entre eux.
- Les règles du discours. Il s'agit de la connaissance des conventions de programmation, qui permettent au programmeur d'acquérir un plus grand sens pour certains aspects du code source et/ou de la documentation. Un exemple est : « Si un nom de variable porte un mot, le sens de ce mot est lié en quelque sorte à l'objectif de cette variable ».

Les principales répercussions de ce modèle concernent la construction, l'utilisation, et la manipulation de la base de connaissances. Les connaissances du domaine sont similaires au domaine de Brooks. Les plans formellement définis concordent aux correspondances entre les différents domaines de bas niveau mentionnés par Brooks. Mais les correspondances entre les domaines de haut niveau ne sont pas abordées par les plans. Ceci est une conséquence directe de la procédure formelle et de la structure rigide de plans. Ils ont acquis leur pouvoir expressif en bas niveau en sacrifiant la puissance nécessaire pour exprimer les relations des domaines de haut niveau.

3.2 Représentation des design patterns

Notre intérêt dans ce document est de s'intéresser à la compréhension des programmes lorsque l'utilisateur utilise une représentation.

Il existe de nombreuses expériences² dans la littérature mettant en avant la compréhension des programmes et les représentations des programmes. La représentation visuelle la plus couramment utilisée pour décrire l'architecture d'un système est celle de diagrammes de classes UML. Les différents travaux réalisés ont analysé, évalué et développé des critères pour rendre la tâche de la compréhension de programme plus facile en utilisant des diagrammes de classes UML.

Cependant, les représentations qui nous intéressent sont celles qui mettent en valeur les données des design pattern. Or, aucune de ces expérimentations ne s'occupe du problème de bien représenter les patrons de conception pour la compréhension des programmes. Ces dernières années, des auteurs se sont intéressés à ce problème et ont élaboré des représentations.

Une étude des différentes représentations a été réalisée par Cepeda Porras [Por08]. Le but de son expérience a donc été de vérifier s'il existe effectivement une diminution de la charge cognitive et donc une amélioration à la compréhension de programmes pendant l'exécution de tâches basiques à la compréhension de patrons de conception [Por08]. L'étude compare quatre représentations :

- La représentation standard, nommée « UML Collaboration Notation » [Vli98]
- La représentation « stereotype-enhanced UML diagrams » de Dong *et al.* [DYZ07]
- La représentation « Pattern-enhanced class diagrams » de Schauer et Keller [SK98]
- La représentation « « pattern :role » notation » de Gamma *et al.* [Gam96]

Les résultats de son étude étaient intéressantes mais manquaient parfois de support ou de profondeur. En effet, son travail se résume principalement à une énumération de conclusions, mais aucune explication n'est donnée afin de donner des renseignements sur « le pourquoi » de ces conclusions. Le but de ce document est d'améliorer cet aspect de l'expérience tout en ajoutant des paramètres supplémentaires.

Dans la suite de ce chapitre, nous allons détailler et illustrer avec un exemple de système de fichier les différents représentations qui ont été utilisées par Cepeda Porras [Por08]. De plus, une nouvelle représentation, nommée « Stereotype Pattern Enhanced », créée par nos soins sera également présentée.

3.2.1 UML Collaboration Notation

Cette représentation est une représentation fortement visuelle. L'avantage d'UML Collaboration Notation [Vli98] est qu'elle peut être utilisée pour n'importe quel diagramme de classes UML. Son principe est d'ajouter des informations des design patterns utilisés dans ce diagramme de classes UML. Dans la suite, nous utiliserons *UMLNotation* pour parler de cette représentation.

Information sur les design patterns

Cette représentation utilise deux techniques pour mettre en avant les informations des design patterns. La première technique consiste à construire des ellipses en pointillés qui portent les noms

2. Les auteurs de ces expériences sont, entre autres, Purchase *et al.*, Eichelberger et Gudenberg, Sun et Wong.

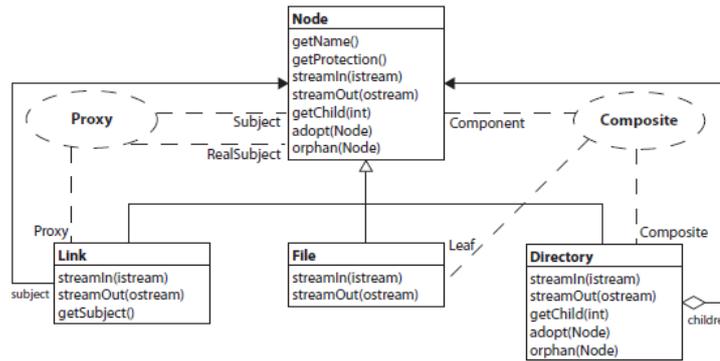


FIGURE 3.3 – Exemple d’une représentation UML Collaboration Notation

des design patterns. Ensuite, cette représentation utilise des lignes, aussi en pointillés, afin de relier les patterns avec les classes participantes. A noter que chacune de ces lignes porte le nom du rôle, du design pattern, joué par la classe liée. Si le diagramme contient deux ou plusieurs instances différentes d’un même design pattern, alors une ellipse sera utilisée pour chacune des instances. En effet, certains design patterns n’acceptent que des rôles uniques, une représentation donnant plusieurs classes pour un rôle unique n’aurait pas de sens. De plus, il serait difficile pour le lecteur de comprendre la structure.

Néanmoins, les notations ajoutées au diagramme (surtout les lignes pointillées) peuvent nuire à la lisibilité du diagramme de façon significative, surtout dans des diagrammes avec un grand nombre de classes.

Si on reprend l’exemple de la figure 3.3, on peut sortir facilement les informations supplémentaires apportées par cette représentation :

- Node joue le rôle de « Component » dans le design pattern « Composite » et le rôle « Subject » et « RealSubject » dans le design pattern « Proxy ».
- Directory joue le rôle de « Composite » dans le design pattern « Composite ».
- File joue le rôle de « Leaf » dans le design pattern « Composite ».
- Link joue le rôle de « Proxy » dans le design pattern « Proxy ».

3.2.2 Stereotype-enhanced UML diagrams

La représentation stereotype-enhanced UML diagrams [DYZ07], nommée plus couramment *Dong* en référence à son auteur, est une représentation essentiellement textuelle. Comme pour la représentation d’UML Collaboration Notation [Vli98], cette représentation peut être utilisée dans n’importe quel diagramme de classes UML. Cependant, vu son caractère textuel, les différentes classes peuvent devenir plus conséquentes et demander, par conséquence, plus d’espace.

Informations des design patterns

Les informations des design patterns utilisées sont symbolisées par du texte. Ces informations textuelles sont ajoutées au diagramme via des stéréotypes. Ces différents stéréotypes sont autorisés dans les diagrammes de classes UML. En ce qui concerne le placement de ces informations, les auteurs ont décidé de les placer devant les noms des classes, des attributs et des méthodes.

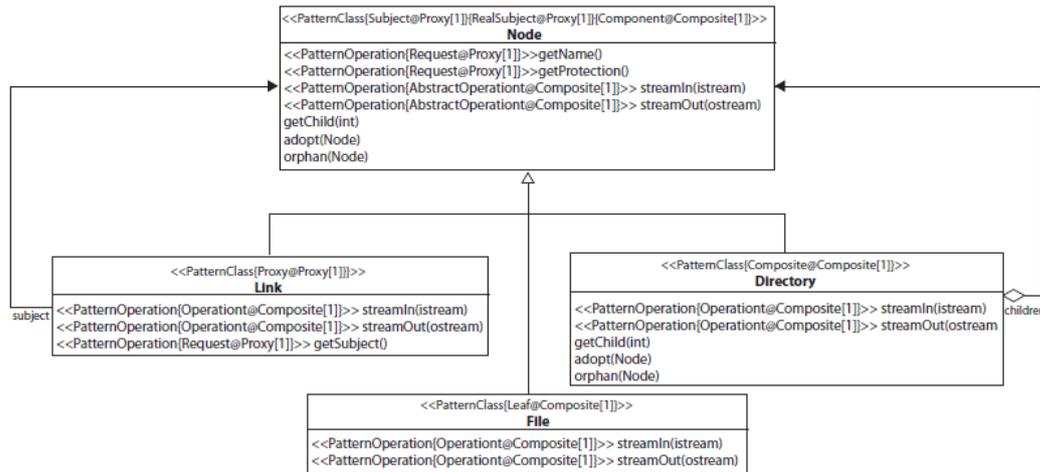


FIGURE 3.4 – Exemple d’une représentation de Dong

Description de l’information textuelle

D’une manière générale, ces stéréotypes indiquent les mêmes informations quelque soit le type d’élément. En effet, mis à part le premier mot qui indique si on a affaire à une classe, une méthode ou un attribut, on retrouve « le rôle, le nom et le numéro d’instance par rapport au design pattern dans lequel la **classe**/l’**attribut**/l’**opération** intervient » :

- Pour les classes : `<< PatternClass { Role@DesignPattern [NuméroInstance] } >>` .
- Pour les attributs : `<< PatternAttribute { Role@DesignPattern [NuméroInstance] } >>` .
- Pour les opérations : `<< PatternOperation { Role@DesignPattern [NuméroInstance] } >>` .

Plus concrètement, la partie « Role@DesignPattern » donne le nom du design pattern (« @Design-Pattern »), dans lequel la classe, l’attribut ou l’opération joue le rôle spécifié par la partie « Role@ ». Tandis que « NuméroInstance » représente le numéro de l’instance du design pattern représenté.

Cette représentation prend aussi en compte si une classe, un attribut ou une méthode joue plusieurs rôles dans différents design patterns, ces informations seront mises les unes à la suite des autres, de la manière suivante :

```
<< PatternClass { Role@DesignPattern1 [NuméroInstance] } { Role@DesignPattern2 [NuméroInstance] } >>
```

Si un design pattern ne possède qu’une seule instance au sein du diagramme, l’information [NuméroInstance] peut être omise.

La figure 3.4, représente un diagramme de classe sous cette représentation. Sans faire une liste exhaustive de toutes les informations que l’on y trouve, nous pouvons en citer quelques unes :

- La classe « Node » joue le rôle de « Component » dans la première instance du design pattern « Composite » et le rôle « Subject » et « RealSubject » dans la première instance du design pattern « Proxy ».
- La méthode « getName() » de la classe « Node » joue le rôle « Request » dans la première instance du design pattern « Proxy ».
- ...

3.2.3 Pattern-enhanced class diagrams

La représentation de Schauer et Keller, appelée Pattern-enhanced class diagrams [SK98], est, contrairement à de celle de Dong [DYZ07], une représentation essentiellement visuelle. Elle diffère donc totalement de la représentation de Dong qui se base sur une représentation textuelle. La base de cette représentation est pourtant la même qu'utilise Dong, et peut donc elle aussi être utilisée dans n'importe quel diagramme de classes UML. Dans un soucis de raccourci, cette représentation est principalement nommée *Schauer*.

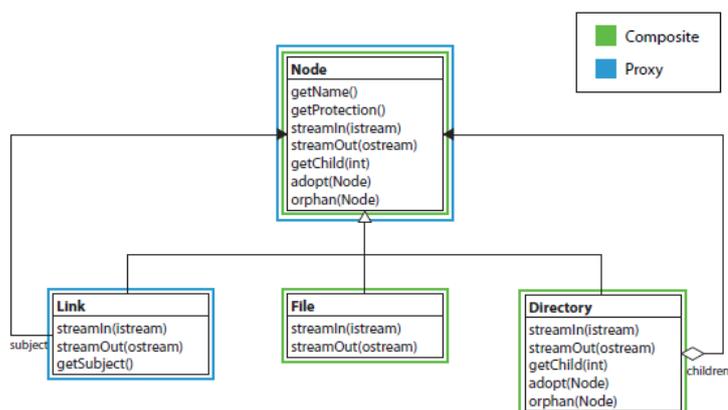


FIGURE 3.5 – Exemple d’une représentation de Schauer

Informations des design patterns

Cette représentation est basée sur une représentation UML. Elle utilise des bordures de différentes couleurs pour entourer l’ensemble des classes présentes sur le diagramme. Ces couleurs indiquent à quel(s) design pattern(s) appartient une classe. En effet, un cadre est ajouté en haut à droite afin d’associer les différentes couleurs à leur design pattern respectif. De plus, cette représentation utilise la représentation canonique afin d’aider les utilisateurs à déduire les rôles que joue chaque classe. Cependant, il n’est pas toujours aisé de ressortir ces informations si le lecteur n’a pas une bonne maîtrise des structures des design patterns.

Le système des bordures colorées permettent d’éviter une surcharge du diagramme due au fait qu’une classe appartienne à de nombreux design patterns en même temps (surcharge présente dans une représentation de Dong [DYZ07] par exemple). Mais comme énoncé ci-dessus, la détection des rôles n’est vraiment pas optimisée ici.

La figure 3.5, représente un diagramme de classe sous cette représentation. Comme cette représentation est essentiellement graphique, la lecture est assez intuitive :

- La classe « Node » appartient aux design patterns « Composite » et « Proxy ».
- La classe « Link » n’appartient qu’au design pattern « Proxy ».
- ...

Par contre, rien n’indique clairement et facilement que la classe « Node » joue le rôle « Request » dans le design pattern « Proxy »...

3.2.4 « Pattern :role » notation

Dernière représentation présente dans l'expérience de Cepeda Porras [Por08], est la représentation « pattern :role » notation créée par Gamma *et al.* [Gam96]. Pour les mêmes motivations que pour les représentations précédentes, celle-ci sera essentiellement nommée *Gamma*. Gamma est une représentation textuelle et visuelle. Utilisant aussi les mêmes bases que les autres représentations, on peut employer cette représentation à partir de n'importe quel digramme de casse UML.

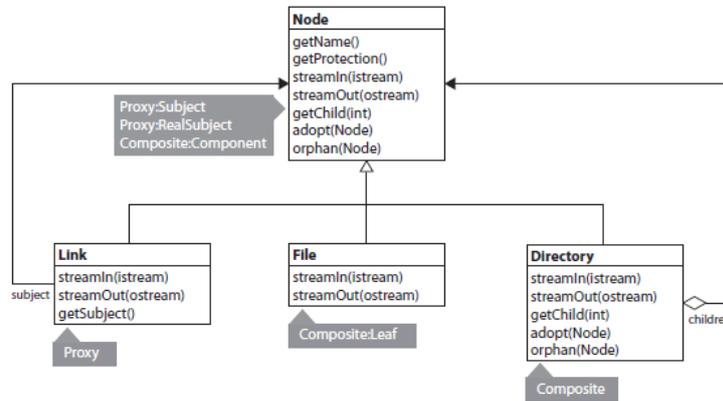


FIGURE 3.6 – Exemple d'une représentation de Gamma

Informations des design patterns

Cette représentation reprend la structure d'une représentation UML mais ajoute des boîtes grises à côté de chaque classe. C'est dans ces différentes boîtes que réside l'intérêt de cette représentation. Le but est de rassembler dans celles-ci l'ensemble des informations sur les design patterns. Chaque classe du diagramme est donc associée à une boîte, située le plus proche possible de cette classe. Ce rapport de proximité évite ainsi toutes ambiguïtés.

Cette représentation est très lisible car elle met les informations liées aux design patterns dans une autre dimension que le reste du schéma. Cependant, cette représentation pourrait augmenter considérablement la taille du diagramme d'origine. En outre, la combinaison de ces boîtes grises sur un fond blanc pourrait conduire à des problèmes de lecture sur des supports imprimés.

La figure 3.6, représente un diagramme de classe sous cette représentation. Nous allons détailler ici, la boîte grise associée à la classe « Node » :

- La classe « Node » joue les rôles de « Subject » et de « RealSubject » dans le design pattern « Proxy ».
- La classe « Node » joue le rôle de « Component » dans le design pattern « Composite ».

3.2.5 Stereotype Pattern Enhanced

Cette représentation inédite est plus particulière dans le sens qu'elle résulte d'une réflexion personnelle de notre part. Cette technique de visualisation de design patterns utilise une représentation visuelle et textuelle. Pour la visualisation des design patterns, cette représentation montre des diagrammes des classes augmentés de patrons de conception, où des contours colorés autour de classes sont utilisés pour identifier quelles classes participent à quel patron. Le lecteur trouvera ici une idée reprise directement de la représentation de Schauer [SK98].

Le seul défaut de cette représentation est, comme nous l'avons décrit, qu'elle ne donne aucune

information en plus sur les rôles des design patterns. C'est cette absence d'information qui nous a donné l'envie de l'améliorer et de la fusionner avec un aspect d'une autre représentation.

Pourquoi reprendre cette représentation précisément et y ajouter des informations ? Ce choix est d'abord motivé par le fait que l'utilisation des couleurs nous semblait pertinente. En effet, Yusuf *et al.* [YKM07] ont mené une étude pour analyser l'utilisation des diagrammes de classes UML pour la compréhension de programme. Ils ont remarqué que l'utilisation d'information additionnelle telle que les stéréotypes et les couleurs rendent les diagrammes plus lisibles et peuvent donc réduire l'effort de compréhension de programme, ce qui est, notre but premier.

Ensuite, nous nous sommes intéressés à l'information à ajouter. Au vu de la faiblesse majeure de Schauer [SK98] à fournir des données sur le rôle joué d'une classe dans un design pattern, nos recherches se sont concentrées sur ce domaine. Les trois autres représentations (UMLNotation [Vli98], Dong [DYZ07] et Gamma [Gam96]) offrent ces informations. L'ajout de ces informations textuelles étant capital quant à l'utilité de cette future représentation, nous avons décidé de choisir celui dont la complexité était la plus faible et donc le plus à même de favoriser notre représentation. Les études couvrant ces représentations étant peu nombreuses, nous nous sommes basés sur une étude réalisée en 2007 qui compare ces trois représentations [DYZ07]. Dans cette étude, la représentation de Dong est présentée comme moins complexe par rapport aux représentations UMLNotation et Gamma. Les métriques utilisées pour faire l'analyse de la complexité sont les suivantes : complexité du graphe, nombre d'arcs et de nœuds, nombre de caractères et d'unités lexicales, la métrique de complexité McCabe, la complexité de diagramme de classes et le nombre d'unités graphiques.

Nous avons donc décidé de reprendre l'idée des couleurs en y ajoutant des informations sur les rôles joués par les différents éléments du diagramme. Dès lors, nous avons rassemblé les avantages de l'aspect visuel de la représentation de Schauer [SK98] et de l'aspect textuel de la représentation de Dong [DYZ07] afin de créer cette représentation.

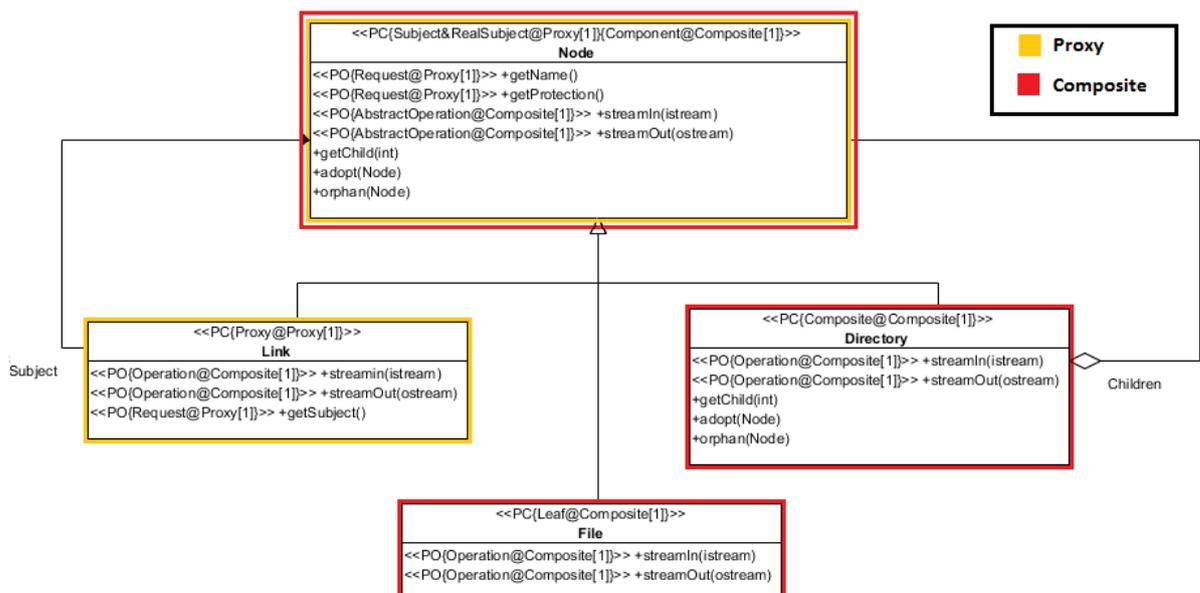


FIGURE 3.7 – Exemple d'une représentation Stereotype Pattern Enhanced

Informations des design patterns

Les informations textuelles suivent les règles définies dans la représentation *Stereotype Enhanced* de Dong [DYZ07]. A l'exception faite que, pour une raison de gain de place, nous avons décidé de remplacer le premier mot composé qui indique s'il s'agit d'une classe, une méthode ou un attribut par son acronyme. Ainsi l'information textuelle peut avoir trois formes différentes :

- pour les classes : $\ll \text{PC} \{ \text{Role@DesignPattern} [\text{NuméroInstance}] \} \gg$.
- pour les attributs : $\ll \text{PA} \{ \text{Role@DesignPattern} [\text{NuméroInstance}] \} \gg$.
- pour les opérations : $\ll \text{PO} \{ \text{Role@DesignPattern} [\text{NuméroInstance}] \} \gg$.

Comme pour la représentation de Dong [DYZ07], si une classe, un attribut ou une méthode joue plusieurs rôles dans différents design patterns, ces informations seront mises les unes à la suite des autres, de la manière suivant :

$\ll \text{PC} \{ \text{Role@DesignPattern1} [\text{NuméroInstance}] \} \{ \text{Role@DesignPattern2} [\text{NuméroInstance}] \} \gg$

De plus, si un design pattern ne possède qu'une seule instance au sein du diagramme, l'information [NuméroInstance] peut être omise.

Cependant, si une classe, un attribut ou une opération joue **au moins deux rôles différents dans une même instance** d'un design pattern, alors l'information PC, PA ou PO prendra la forme suivante :

$\ll \text{PC} \{ \text{Role1\& Role2 @ DesignPattern} [\text{NuméroInstance}] \} \gg$. On évite ainsi de répéter inutilement le nom du design pattern.

En ce qui concerne les informations visuelles, nous avons repris l'idée d'entourer les différentes classes de couleurs. Les couleurs sont utilisées pour spécifier l'appartenance des classes aux différents design patterns. Une légende est présente dans le coin supérieur droit de la représentation pour spécifier la couleur associée au design pattern.

Nous avons donc une représentation se basant essentiellement sur celle de Dong [DYZ07] mais avec un ajout de couleurs. Cette représentation, illustrée dans un exemple par la figure 3.7 peut être comprise telle que ci dessous :

- Nous pouvons voir via la couleur, que la classe "Directory" joue un rôle dans le design pattern "Composite".
- Nous pouvons voir via l'information textuelle, que la classe "Node" joue les rôles de "Subject" et "RealSubject" dans la première instance du design pattern "Proxy" et aussi qu'elle joue le rôle de "Component" dans la première instance du design pattern "Composite".
- ...

Science de la vision

La science de la vision est dédiée à l'étude interdisciplinaire du système et de la perception visuelle. Au sein de l'informatique, cette science s'intéresse à la compréhension du système de vision humaine pour évaluer la charge cognitive de certaines tâches ou pour détecter les zones d'attention. La section 4.1 présente le lien entre le processus cognitif des êtres humains et la théorie de la vision. Les différents mouvements oculaires qui sont produits lors de la vision d'un stimuli par un sujet sont explicités au sein de la section 4.2. Enfin, la section 4.3 décrit la technique de l'oculométrie, les mesures qui en découlent et son utilisation au sein du génie logiciel.

Ce chapitre s'inspire globalement des ouvrages de Palmer [Pal99] et de De Smet et Lempereur [DL11].

Sommaire

4.1	Processus cognitif et théorie de la vision	30
4.2	Catégorisation des mouvements oculaires	30
4.3	Oculométrie	32
4.3.1	L'histoire de l'oculométrie	32
4.3.2	Fonctionnement d'un oculomètre	32
4.3.3	Mesures oculométriques	33
4.3.4	Utilisation de l'oculométrie dans le génie logiciel	37

4.1 Processus cognitif et théorie de la vision

L'existence d'un lien entre la théorie de la vision et les processus cognitifs des individus humains nous permet d'évaluer la charge de travail d'un être humain à partir des données récoltées par un système oculométrique. D'après Palmer [Pal99], nous déplaçons nos yeux dans le but de mettre en haute résolution une portion particulière de notre champ de vision. La *fovéa* constitue la zone de meilleure acuité visuelle de l'être humain. Elle se situe au centre de la rétine et a une taille d'angle visuel d'environ 1°. Un individu humain ne peut voir très nettement que ce qui se trouve sur sa fovéa. Le plus souvent, nous détournons notre attention sur ce point, afin de nous concentrer sur l'objet d'intérêt. En suivant les mouvements des yeux d'un observateur, il est possible de déterminer ce qu'il trouve intéressant, c'est-à-dire ce qui attire son attention [Pal99].

Une contribution inestimable de Yarbus [TWK+10] réside dans une expérience qu'il mit en place en 1967. Cette expérience montre que le chemin visuel d'une personne regardant une image dépend de la tâche qu'elle doit accomplir. Au cours de celle-ci, il demanda à un même sujet de regarder la peinture « The Unexpected Visitor » sept fois. Une instruction différente était annoncée à la personne avant chaque observation : (a) observation de l'image sans instruction, (b) estimer le revenu économique des personnages de l'image, (c) estimer l'âge des personnages, (d) présumer ce que les personnes faisaient avant l'arrivée du « visiteur inattendu », (e) se rappeler des vêtements porté par les personnages, (f) se souvenir des positions des personnes et des objets dans la pièce, (g) estimer depuis combien de temps le « visiteur inattendu » n'était plus venu dans la famille. Les données résultantes (présentées au sein de la figure 4.1) de ces expériences illustrent de manière convaincante que l'altération des instructions données à l'observateur (c'est-à-dire leur tâche pendant la vision) a un effet profond sur son comportement d'inspection. Yarbus observa lui-même :

« En fonction de la tâche qu'un individu doit réaliser, [..], la distribution de ses points de fixations sur un objet varie de manière correspondante, en raison de la disposition des éléments de l'information à des endroits différents dans l'objet. » [Yar67, p. 192]

Par conséquent, à partir d'une tâche donnée, il est possible d'estimer si un individu regarde aux bons endroits d'un diagramme. Wade *et al.* [TWK+10] relatent également une autre expérience¹ de Yarbus qui montre que l'attention est fonction de l'individu.

4.2 Catégorisation des mouvements oculaires

Les mouvements oculaires permettent aux êtres humains de saisir une image et d'assurer sa stabilité sur la rétine. La littérature distingue 2 grands types de mouvements oculaires :

1. les mouvements de stabilisation de l'image.
2. les mouvements d'orientation du regard.

Les premiers comprennent les mouvements vestibulo-oculaires et optocinétiques. Leur but est de fixer le regard sur une cible lorsque la tête bouge. Les deuxièmes mouvements ont pour fonction de garder la fovéa sur une cible même lorsque celle-ci se déplace. Pour ce faire, notre système visuel produit des fixations, des saccades, des poursuites, ou de vergences.[RT05]

Fixations : une fixation est un moment pendant lequel les yeux sont relativement stationnaires pour prendre ou encoder de l'information. L'objet de l'attention reste en position et permet à celui-ci,

1. Yarbus demanda à plusieurs personnes de visionner une peinture sans instruction particulière. Il releva que les chemins visuelles étaient similaires, mais pas identiques.

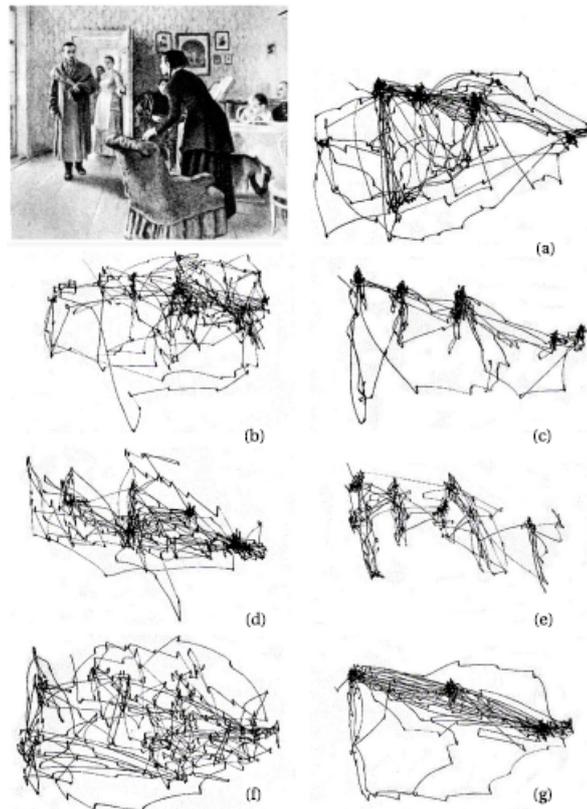


FIGURE 4.1 – Observation d’une image suivant différentes questions posées [TWK+10].

lorsqu’il est stable, de rester sur la fovéa. Lorsque nous lisons ou que nous cherchons après un objet, nos yeux restent relativement stationnaires pendant les fixations. Celle-ci durent au minimum 100 à 150 millisecondes, mais ont une moyenne de 200 à 300 millisecondes [Ray98].

Saccades : la saccade est un mouvement de l’œil rapide qui provient entre deux fixations. Aucun encodage et très peu ou pas du tout de traitement se produisent durant une saccade. Bien que nous ayons l’impression que nous puissions traiter tout le champ visuel dans une fixation simple, en réalité, nous serions incapables de traiter l’information en dehors de la vision fovéale si nous étions incapables de déplacer nos yeux [RC07]. Les saccades ramènent l’image d’un nouvel objet sur la fovéa. Elle est le mouvement le plus rapide dont nous sommes capables, une fois commencé, on ne peut l’arrêter [RT05].

Poursuites : elle garde l’image d’un objet en mouvement sur la fovéa. La poursuite est un comportement oculaire qui est déclenché volontairement et qui requiert un stimulus en mouvement [RT05]. Si la cible se déplace par saut, des saccades seront produites. Par contre si la cible se déplace de manière continue, l’individu produira des poursuites lentes.

Vergences : ils ajustent l’orientation des yeux en fonction de la distance de profondeur de l’objet de l’attention. Lorsque nous regardons un objet qui s’éloigne de nous, chaque œil tourne dans une direction opposée afin de maintenir l’image de cet objet alignée sur la fovéa de chaque œil [RT05].

Mouvements vestibulo-oculaires et optocinétiques : ils utilisent une entrée visuelle pour maintenir les images stables sur la rétine durant des rotations de la tête ou des déplacements du corps

(brefs et rapides ou lents et prolongés).

4.3 Oculométrie

L'oculométrie est une technique à partir de laquelle les mouvements oculaires d'un individu sont mesurés. Ces mesures permettent de connaître les endroits sur lesquels le sujet a posé son regard et le séquençage des déplacements du regard [PB06]. Le suivi des mouvements oculaires des personnes peut aider les chercheurs d'interaction homme-machine à comprendre le traitement visuel qui est opéré lors de l'utilisation d'interface système. Cette technique est également utilisée pour transposer les mouvements oculaires en signaux de commande afin de permettre à certains individus d'interagir avec des interfaces sans la nécessité d'une souris ou d'un clavier.

4.3.1 L'histoire de l'oculométrie

Les premières descriptions qualitative des mouvements oculaires datent de plus d'un siècle. A la fin du 19^e siècle, Ahrens, Delabarre et Huey sont les premiers à essayer d'enregistrer les mouvements oculaires. Leur procédé consistait à transférer les mouvements sur une surface couverte de suie par des petits leviers fixés sur le globe oculaire [Dre10].

Au fil des années, d'autres méthodes ont été mises en place pour enregistrer le regard. Une technique électro-oculographique, par exemple, a utilisé des électrodes montées sur la peau autour de l'œil. Ainsi les différences de potentiel électrique pouvaient être mesurées de manière à détecter les mouvements oculaires [PB06]. Ces approches s'avéraient être fort invasives. La plupart des systèmes oculométriques modernes utilisent maintenant des images vidéos de l'œil pour déterminer où un sujet regarde.

4.3.2 Fonctionnement d'un oculomètre

Les oculomètres modernes mesurent l'endroit où le regard est posé à partir de la méthode « réflexion de la cornée / centre de la pupille ». Ce type d'*Eye Tracker* utilise un ordinateur accompagné d'une caméra infra-rouge montée sur (ou à côté) de l'écran. L'ordinateur possède un programme de traitement d'image qui localise et identifie les caractéristiques de l'œil durant le suivi. La lumière infra-rouge de la caméra est projetée dans l'œil pour créer de fortes réflexions et rendre le suivi plus facile (l'infra-rouge permet d'éviter les reflets de la lumière naturelle). La lumière rentre dans la rétine et une grande proportion de celle-ci est réfléchi en retour, faisant ainsi apparaître la pupille sous la forme d'un disque de lumière bien défini. Le reflet cornéen est aussi produit par la lumière infra-rouge : il apparaît comme un petit reflet, de forte intensité.

Dès que le programme de traitement d'image a identifié le centre de la pupille et la réflexion de la cornée, il est en mesure de calculer le point du regard (sur base de différents calculs trigonométriques). Comme nous pouvons le voir sur la figure 4.2, il est également possible de déterminer approximativement, à la main, le point du regard sur base de la position de la réflexion cornéenne.

Le matériel utilisant ce système de vidéo nécessite d'être finement configuré pour chaque personne dont le regard sera mesuré. L'étape de *calibration* exécute cette configuration. Cette étape enregistre pour différents points affichés à l'écran : les coordonnées (x,y) du point de l'écran et le rapport « centre de la pupille / réflexion de la cornée ».

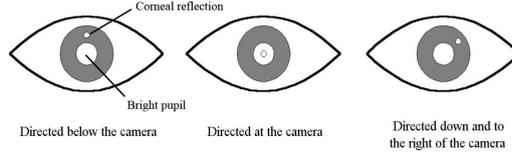


FIGURE 4.2 – Position de la réflexion cornéenne modifiée selon le point du regard [RL01].

4.3.3 Mesures oculométriques

Cette section a pour but de présenter une majorité des mesures oculométriques qu'il est possible de trouver dans la littérature. Ces mesures se font sur base des données récoltées (fixations et saccades) par des oculomètres classiques. La taille de la pupille ou le taux de clignements des yeux peuvent aussi être calculés, mais ils ne seront pas présentés dans ce document. L'objectif de cette partie n'est pas d'être exhaustif, un résumé des mesures les plus souvent utilisées peut être trouvé dans [PB06]. Cette section s'inspire principalement des ouvrages [DL11] et [PB06].

Avant de présenter les métriques, il est nécessaire de définir les différentes notations qui seront utilisées dans cette section.

A : ensemble de réponses (A_j = ensemble de réponses pour la question j).

F : ensemble de fixations.

S : ensemble de saccades.

$AORI_i$: ensemble des zones d'intérêt pertinentes pour la question i .

$AOII_i$: ensemble des zones d'intérêt non-pertinentes pour la question i .

AOI_i : ensemble de zones d'intérêt pour la question i ($AOI_i = AORI_i \cup AOII_i$).

$tF : F \rightarrow \mathbf{IN}$: La durée d'une fixation spécifiée (en millisecondes).

$tS : S \rightarrow \mathbf{IN}$: La durée d'une saccade spécifiée (en millisecondes).

$fix : AOI \times A \rightarrow F^*$: Les fixations dans une zone d'intérêt spécifique pour une réponse particulière.

$surface : AOI \rightarrow \mathbf{IN}$: La surface de la zone d'intérêt spécifiée (en pixels²).

$départ : S \rightarrow F$: La fixation de départ d'une saccade.

$fin : S \rightarrow F$: La fixation de fin d'une saccade.

$FAORI_i$: L'ensemble des fixations dans une zone d'intérêt pertinente pour la réponse i ($i \in A_j$) tel que :

$$FAORI_i = \bigcup_{k \in AORI_j} fix(k, i).$$

$FAOII_i$: L'ensemble des fixations dans une zone d'intérêt non-pertinente pour la réponse i ($i \in A_j$) tel que :

$$FAOII_i = \bigcup_{k \in AOII_j} fix(k, i).$$

$FAOI_i$: L'ensemble des fixations dans une zone d'intérêt (pertinente, non-pertinente ou nulle) pour la réponse i ($i \in A_j$) tel que :

$$FAOI_i = \bigcup_{k \in AOI_j} fix(k, i).$$

$TAORI_i$: Durée totale (en millisecondes) des fixations dans les zone d'intérêt pertinentes pour la réponse i ($i \in A_j$) :

$$TAORI_i = \sum_{f \in FAORI_i} tF(f).$$

$TAOII_i$: Durée totale (en millisecondes) des fixations dans les zones d'intérêt non-pertinentes pour la réponse i ($i \in A$) :

$$TAOII_i = \sum_{f \in FAOII_i} tF(f).$$

$TAOI_i$: Durée totale (en millisecondes) des fixations dans les zone d'intérêt (pertinentes et non-pertinentes) pour la réponse i ($i \in A$) :

$$TAOI_i = TAORI_i + TAOII_i.$$

$\#F_i$: Le nombre total de fixations pour une réponse i d'un sujet ($i \in A$).

$AORIT_oAOII_i$: ensemble de saccades dont la fixation de départ est comprise dans une zone d'intérêt pertinente et la fixation d'arrivée est comprise dans une zone d'intérêt non pertinente pour une réponse i d'un sujet ($i \in A$) :

$$AORIT_oAOII_i = \bigcup_{s \in S_i} s.$$

tel que $départ(s) \in FAORI_i$ et $fin(s) \in FAOII_i$

$AOIT_oAOI_i$: ensemble de saccades dont les fixations de départ et d'arrivée sont toutes les deux comprises dans une zone d'intérêt (pertinente, non pertinente ou null) pour une réponse i d'un sujet ($i \in A$) :

$$AOIT_oAOI_i = \bigcup_{s \in S_i} s.$$

tel que $départ(s) \in FAOI_i$ et $fin(s) \in FAOI_i$

Zone d'intérêt Une zone d'intérêt est une zone visuelle caractérisée par un certain niveau d'intérêt. Elle est considérée par l'expérimentateur comme pertinente (intérêt élevé) pour une tâche si le sujet doit regarder cette zone pour répondre à la tâche. Une zone peut également être non-pertinente (intérêt faible) ou bien peut être ignorée (le cas, par exemple, des zones se situant en dehors de la donnée visuelle).

Sur base de ces notations, nous allons maintenant décrire plusieurs métriques.

Enveloppe convexe Comme nous pouvons le voir sur la figure 4.3, pour un ensemble de point P dans un plan à deux dimensions, l'enveloppe convexe est l'ensemble convexe le plus petit qui contient l'ensemble des points P . Si nous considérons les fixations comme des points, l'enveloppe convexe permet de mettre en évidence le parcours visuel d'un sujet. Plus la surface de l'enveloppe est faible, plus le sujet a effectué une recherche efficace.

Nombre de fixation par zone d'intérêt Un nombre élevé de fixation dans une zone d'intérêt indique qu'elle est plus remarquable ou plus importante pour la personne que les autres zones [PBP04].

Durée moyenne des fixations Une durée de fixation plus longue peut indiquer deux phénomènes différents [JC76] :

1. Le sujet a du mal à extraire l'information.

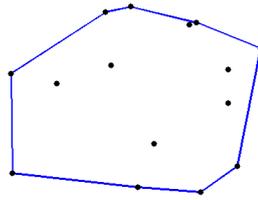


FIGURE 4.3 – Enveloppe convexe

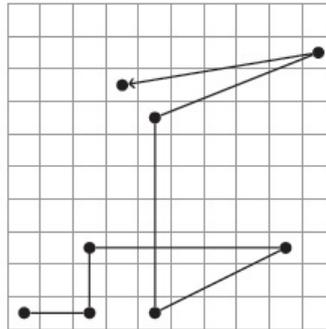


FIGURE 4.4 – Exemple de densité spatiale égale à 8% [DL11]

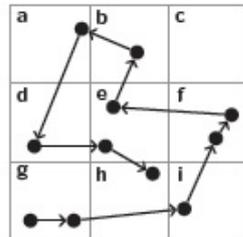


FIGURE 4.5 – Exemple de parcours visuel [DL11]

	a	b	c	d	e	f	g	h	i
a				1					
b	1								
c									
d					1				
e		1						1	
f					1	1			
g							1		1
h									
i						1			

Tableau 4.1 – Matrice de transition de la figure 4.5 [DL11]

2. Le sujet est plus attiré par l'objet.

Densité spatiale La figure 4.4 représente un exemple d'une densité spatiale d'un chemin visuel égale à 8%. Cette métrique est mesurée sur base de la distribution spatiale des fixations. L'image présentée au sujet est divisée en zone selon une grille. La densité spatiale est égale au nombre de cellules dans lesquelles il y a au moins une fixation, divisé par le nombre de cellules totales de la grille. Si la densité est faible, le sujet a focalisé son regard dans une petite zone, il a donc procédé à une recherche concentrée et efficace [PB06].

Matrice de transition La matrice de transition intègre les transitions du regard d'une région à une autre. Le tableau 4.1 illustre la matrice de transition du parcours visuel présent au sein de la figure 4.5. Les valeurs des cellules de la matrice sont restreintes à 1 ou 0. Si une cellule (x,y) possède la valeur 1, alors il existe, dans le parcours visuel, une transition de la cellule x à la cellule y. Deux chemins visuels avec les mêmes enveloppes convexes et les mêmes densités

spatiales peuvent avoir une matrice de transition différente.

Densité de transition La densité de transition est calculée sur base de la matrice de transition, de la manière suivante :

$$TRANSITION_DENSITY = \frac{\sum_{x \in C} isFilled(x)}{\#C}$$

où C est l'ensemble des cellules dans la matrice de transition, $\#$ retourne la cardinalité d'un ensemble et $isFilled : C \rightarrow \{0, 1\}$ retourne 1 si la cellule spécifiée est remplie et 0 sinon. Une densité de transition faible indique des transitions directes, alors qu'une densité plus élevée révèle un va-et-vient continu et donc une incertitude de la part du sujet.

Pourcentage de transition AORI vers AOII Le pourcentage de transition d'une zone d'intérêt pertinente vers une zone d'intérêt non pertinente pour une réponse i est calculée à partir de la fonction suivante :

$$PERCENTAGE_TRANSITION_AORI_TO_AOII_i = \frac{\#AORIT_oAOII_i}{\#AOIT_oAOI_i}$$

Nous avons mis en place cette métrique afin de pouvoir juger l'importance d'interprétation des zones d'intérêt pertinentes. Une pourcentage de transition AORI vers AOII élevé indique une mauvaise interprétation des zones d'intérêt pertinentes.

Ratio de la durée des fixations et des saccades Le ratio de la durée des fixations et des saccades pour une réponse i est calculée à partir de la fonction suivante :

$$FIXATION_SACCADES_DURATION_RATIO_i = \frac{\sum_{f \in F_i} tF(f)}{\sum_{s \in S_i} tS(s)}$$

Cette métrique compare le temps consacré au traitement (fixations) et le temps consacré à la recherche (saccades). Un ratio élevé de cette mesure indique plus de traitement ou moins de recherche.

Taux normalisé de fixations pertinentes Cette métrique a été suggérée par Jeanmart [Jean08]. Elle est définie comme le ratio entre le nombre normalisé de fixations dans les zones d'intérêt pertinentes et le nombre normalisé de fixations dans les zones d'intérêt non pertinentes. Un taux élevé correspond à un sujet réalisant peu d'effort et un taux faible correspond à un sujet réalisant beaucoup d'effort. Cette métrique est une version simplifiée de la métrique *IN AORI / IN AOII normalisé* détaillée ci-dessous.

IN AORI / IN AOII normalisé (par zone d'intérêt et par surface) De Smet et Lempereur [DL11] proposent cette métrique dérivée de la métrique « fixations par zone d'intérêt normalisé par la taille du texte » [PB06]. Elle mesure la pertinence de l'effort visuel d'un sujet. Elle est basée sur le principe du taux normalisé de fixations pertinentes présenté par Jeanmart [Jean08]. Elle est le ratio entre le nombre de fixations dans les zones d'intérêt pertinentes (AORI) et le nombre de fixations dans les zones d'intérêt non-pertinentes (AOII). Ces valeurs sont normalisées en fonction du nombre de zones d'intérêt **mais également en fonction de leur surface**. Cette normalisation se justifie par le fait qu'il est possible que les zones d'un certain type soient plus nombreuses que celles d'un autre type. De plus, la surface d'une zone sur le diagramme peut influencer le nombre de fixations tombant à l'intérieur de celle-ci. Ce rapport est un indicateur

de la pertinence de l'effort. Le ratio $IN_OUT_Norm_i$ pour la réponse d'un sujet i à une question k est défini tel que :

$$IN_OUT_Norm_i = \frac{\frac{FAORIS_i}{\#AORI_k}}{\frac{FAOIS_i}{\#AOII_k}}$$

où $FAORIS_i$ est le nombre de fixations dans les zones d'intérêt pertinentes normalisé par la surface pour la réponse i :

$$FAORIS_i = \sum_{j \in AORI_k} \frac{\#fix(j, i)}{surface(j)}$$

et où $FAOIS_i$ est le nombre de fixations dans les zones d'intérêt non-pertinentes normalisé par la surface pour la réponse i :

$$FAOIS_i = \sum_{j \in AOII_k} \frac{\#fix(j, i)}{surface(j)}$$

Nombre moyen de fixations post-AORI Cette métrique est inspirée de la métrique « Fixations Post-cible » proposée par Goldberg et Kotval [GK99]. Notre métrique calcule la moyenne du nombre de fixations apparaissant après la visualisation d'une zone d'intérêt pertinente. Le calcul, pour la réponse i est représenté par la fonction suivante :

$$POST_AORI_FIXATIONS_i = \frac{\sum_{f \in FAORI_i} nextFix(f)}{\#AORI_i}$$

où $nextFix(f)$ est le nombre de fixations suivant la fixation f .

4.3.4 Utilisation de l'oculométrie dans le génie logiciel

Depuis plusieurs années, l'oculométrie est utilisée à des fins expérimentales dans plusieurs domaines comme la publicité, la science, l'aviation ou encore la conduite automobile. L'usage de matériel oculométrique est également présent dans des expériences en génie logiciel. Nous allons donc présenter quelques études empiriques réalisées via ce type de matériel.

Dans leur ouvrage [BT04], Bedranik et Tukiainen réalisent une comparaison entre deux outils de suivi de l'attention visuelle : un oculomètre classique et un RFV (*Restricted focus viewer*). Le RFV brouille le stimuli afin de simuler certaines caractéristiques de la vision humaine. L'utilisateur contrôle la portion de l'écran restant visible à partir d'une souris. 18 sujets ont participé à l'expérience. Leurs tâches consistaient à corriger les bogues de trois applications Java. Les résultats montrent que la performance de débogage et la distribution du temps passé sur les zones d'intérêts ne sont pas influencées par RFV, malgré que le comportement dynamique de programmation soit différent. Le nombre de transitions entre les zones d'intérêts lors de l'utilisation de RFV est significativement différent avec l'oculomètre classique. Les auteurs maintiennent donc que le RFV doit être utilisé avec prudence en cas de mesures des transitions dans l'attention visuelle.

Lors d'une autre expérience, Bednarik et Tukiainen [BT06] soulignent le fait que les matériels de suivi du regard n'ont pas encore été fortement employés pour l'étude des aspects comportementaux de la programmation. Ils présentent donc une expérience dans laquelle la compréhension des programmes est investiguée avec l'aide d'un oculomètre. Des programmeurs débutants et intermédiaires ont utilisé un programme de visualisation pour aider leur compréhension, tandis que les informations sur leurs fixations et leurs transitions de regard étaient enregistrées. Au sein de leur ouvrage, les auteurs proposent une approche pour étudier les tendances d'observations basées sur les données oculométriques. Après avoir appliqué leur approche, ils concluent que l'oculométrie peut être une source précieuse dans l'analyse des processus cognitifs de la compréhension des programmes.

Un oculomètre est également utilisé lors d'une expérience par Guéhéneuc [Gué06]. Les ingénieurs logiciels utilisent leur vue pour acquérir des informations sur des programmes, habituellement à partir du code source ou de diagrammes de classe UML. Il propose donc de collecter les données oculométriques de sujets lorsqu'ils réalisent des tâches de compréhension de programmes à partir de diagrammes de classe. Son étude révèle l'importance des classes et des interfaces, mais aussi le fait que les sujets semblent peu utiliser les relations binaires comme l'héritage ou la composition.

Comme Guéhéneuc, Yusuf *et al.* [YKM07] mènent une expérience utilisant du matériel de suivi du regard pour évaluer la façon dont un sujet comprend un diagramme de classe UML. Le but de l'étude est d'identifier des caractéristiques spécifiques des diagrammes de classe UML, tels que la disposition, les couleurs et l'utilisation des stéréotypes qui sont les plus efficaces pour soutenir une tâche de compréhension de programme. Les résultats indiquent une variation dans les mouvements oculaires des sujets suivant leur expertise UML et leur capacité à résoudre une tâche de conception logiciel. Les résultats démontrent également une meilleure efficacité des dispositions avec des informations supplémentaires telles que les couleurs ou les stéréotypes lors de tâche de compréhension de programme.

Jeanmart [Jean08] évalue l'impact du patron de conception Visiteur sur la compréhension et la maintenance de programmes orientés-objet. Le design pattern est représenté dans des diagrammes de classe UML à partir de trois programmes *open source* différents. L'étude dévoile une influence significative du patron pour les tâches de modification. En contrepartie, ce dernier n'a pas d'impact significatif sur la compréhension des diagrammes de classe par les sujets.

Cepeda Porras [Por08] dirige une expérience en vue d'évaluer trois différentes techniques de visualisation de patron de conception dans des diagrammes de classe UML. L'étude empirique utilise un oculomètre pour mesurer l'effort des développeurs lors de l'exécution de tâches relatives à la compréhension des patrons de conception (exemple : identifier les compositions, les rôles et les participations). Les données collectées et les analyses montrent que la représentation *stereotype-enhanced* est plus efficace pour identifier les compositions et les rôles, tandis que la technique classique (*UML collaboration notation*) est plus efficace pour localiser les classes participant à un patron de conception.

Van Den Plas [Vdp09] étudie l'influence des patrons de conception Composite et Observateur sur des tâches de compréhension et de modification de diagrammes de classe UML. Les projets

open source JUnit², QuickUML³ et ArgoUML⁴ sont utilisés lors de l'expérience. L'utilisation d'un oculomètre a permis d'analyser l'efficacité de la recherche et le niveau de concentration des sujets. Toutefois, l'étape d'analyse n'a pas fourni de résultat significatif.

La dernière étude que nous présentons est celle de De Smet et Lempereur [DL11]. Contrairement à Jeanmart [Jean08] et Van Den Plas [Vdp09], ceux-ci axent leur étude sur l'impact du patron architectural MVC et de ses variantes sur la maintenabilité interne des programmes. Aucune influence du patron architectural ne ressort de leurs analyses. Par contre, les résultats manifestent une influence du niveau d'étude (Master ou Doctorant) significative sur plusieurs variables oculométriques. Les analyses montrent que les Masters prennent moins de temps pour répondre, observent des espaces plus réduits et se comportent d'une manière plus homogène que les Doctorants.

2. <http://www.junit.org/>

3. <http://sourceforge.net/projects/quj/>

4. <http://argouml.tigris.org/>

Mesure de l'effort de compréhension

Toute réalisation d'une tâche nécessite un effort physique ou mental. Pour une tâche de compréhension de programme, l'effort est essentiellement mental. La charge mentale de travail nécessaire à la réalisation d'une tâche de compréhension de programme est un indicateur très intéressant pour évaluer les outils utilisés. Différentes méthodes existantes permettent de mesurer cette charge mentale. Ce chapitre a pour but de présenter la théorie relative aux mesures de la charge mentale de travail.

La section 5.1 définit le concept de charge mentale de travail. Ensuite, les différentes méthodes existantes pour évaluer ce concept sont présentées en section 5.2. Enfin, la section 5.3 illustre une méthode d'évaluation subjective de la charge mentale, en présentant l'outil NASA-TLX.

Ce chapitre est fortement inspiré des travaux de Stanton [Sta05], de Jiménez [Jim07] et de Cain [Cai07].

Sommaire

5.1	La charge mentale de travail	42
5.2	Évaluation de la charge mentale de travail	42
5.2.1	Mesures de la performance	42
5.2.2	Mesures physiologiques	43
5.2.3	Mesures subjectives	43
5.3	NASA-TLX	43

5.1 La charge mentale de travail

Wilson *et al.* définissent la *charge mentale de travail* comme la partie de la capacité de traitement ou des ressources nécessaires pour répondre aux exigences d'un système [Cai07]. En réalité, la littérature contient de nombreuses définitions, car elle résulte de l'agrégation de nombreuses exigences différentes. C'est pourquoi il est si difficile de la définir de façon unique. En d'autres termes, la charge mentale de travail est un concept hypothétique. Ce concept représente le coût cognitif d'exécution d'une tâche par un être humain. La charge mentale de travail est centrée sur l'être humain et non sur la tâche. Ceci se justifie par le fait qu'il s'agit d'une mesure subjective qui traduit l'influence de facteurs divers sur l'élaboration de la tâche.

Malgré la diversité des opinions sur le concept, la charge mentale de travail est mesurable. La principale raison de mesurer la charge mentale est de pouvoir quantifier le coût mental d'exécution des tâches, en vue de prédire le comportement du sujet et les performances du système. En tant que tel, ce concept représente une mesure provisoire. Celui-ci devrait donner un aperçu des exigences de la tâche où l'augmentation de la charge mentale pourrait conduire à des performances inacceptables. On reconnaît ainsi implicitement que la diminution et l'augmentation de la charge mentale de travail influence la performance, le temps de réponses, le nombre d'erreurs, la variabilité de contrôle, le nombre de tâches effectuées par unité de temps et la capacité résiduelle à traiter d'autres tâches.

5.2 Évaluation de la charge mentale de travail

La mesure de la charge mentale de travail est réalisée dans de nombreux domaines. Cette mesure peut être nécessaire à l'évaluation de nouvelles interfaces utilisateurs, par exemple pour la certification de nouveaux modèles de cockpit d'avions. Son évaluation demande l'utilisation de plusieurs méthodes : la mesure de la performance (tâche primaire ou secondaire), les mesures physiologiques (variation du rythme cardiaque, mouvements des yeux) et les mesures subjectives (SWAT, NASA-TLX).

5.2.1 Mesures de la performance

La mesure de performance peut être classifiée en deux grandes catégories : les mesures de la tâche primaire et les mesures de la tâche secondaire. Dans la plupart des études, la performance de la tâche primaire représente le cœur de l'étude tandis que la méthode de la tâche secondaire peut n'avoir aucune importance pratique et ne sert, en général, qu'à évaluer la charge du sujet. Néanmoins, le niveau de performance du sujet ne fournit aucune information sur les coûts du stress impliqués dans la réponse. Ainsi, alors que la mesure de la performance peut être considérée comme une mesure nécessaire à la charge mentale de travail, elle ne devrait pas être considéré comme suffisante.

La mesure de la tâche primaire tente d'évaluer la performance du sujet sur la tâche d'intérêt. Ceci est utile lorsque la demande excède la capacité du sujet de telle sorte que les performances se dégradent par rapport au niveau de base. Les mesures utilisées pour la tâche primaire sont souvent : la vitesse, la précision, le temps de réaction ou de réponses et le taux d'erreurs.

La mesure de la tâche secondaire fournit un indice sur la capacité restante du sujet lorsqu'il effectue une tâche primaire. Il s'agit donc surtout d'une mesure de diagnostic. Le choix de la tâche secondaire peut avoir un effet profond sur la performance. Certaines tâches secondaires peuvent

également produire un changement indésirable sur la stratégie du sujet. Se souvenir d'un résultat précédemment obtenu ou exécuter une opération arithmétique sont des exemples de tâches secondaires.

5.2.2 Mesures physiologiques

Le but des mesures physiologiques dans l'évaluation de la charge mentale est de mettre au point des mesures ayant des propriétés bien connues qui peuvent être appliquées dans des situations spécifiques. Ce but provient de la nature complexe de la construction de la charge mentale de travail. Ces mesures tentent d'interpréter les aspects physiques qui pourraient être affectés par une variation de la charge mentale de travail. Ces mesures peuvent se baser sur le rythme cardiaque, les mouvements oculaires ou encore sur l'activité du cerveau. Par ailleurs, les aspects physiques, qui sont à la base de ces mesures, nécessitent l'utilisation d'outils technologiques qui coûtent cher.

L'étude contenue dans ce mémoire se base sur des mesures oculométriques. Une présentation de cette technologie est présente au sein de la section 4.3. De plus, la section 4.3.3 contient une définition des mesures oculométriques présentes dans la littérature.

5.2.3 Mesures subjectives

Les mesures subjectives peuvent être effectuées après l'exécution de la tâche principale. Ceci a pour avantage de ne pas perturber l'accomplissement de la tâche par le sujet. La charge mentale de travail est évaluée par le sujet lui-même, sous forme d'un *feedback*. En général, ces auto-évaluations se présentent sous la forme d'échelles numériques ou de graphiques que le sujet doit remplir. Les techniques utilisées sont souvent faciles à comprendre et à utiliser, et sont généralement bien acceptées par les sujets.

Nous pouvons retrouver près de neuf méthodes d'estimation de la charge mentale, parmi les instruments unidimensionnels : l'échelle Cooper-Harper, l'échelle Bedford, l'échelle de Charge globale (Overall Workload), l'échelle de l'Université de Stockholm, l'estimation des magnitudes et la comparaison binaire. D'un autre côté, les instruments de mesure multidimensionnels sont au nombre de trois : SWAT (Subjective Workload Assessment Technique), NASA-TLX (Task Load Index) et le profil de charge mentale WP (Workload Profile).

5.3 NASA-TLX

Le NASA-TLX (Task Load Index) est un outil qui permet de recueillir les évaluations subjectives de charge de travail des sujets. Le NASA-TLX a été développé par Hart et Staveland [HS88, p.139-183]. Comme son nom l'indique, ce test est celui utilisé par la NASA lorsqu'ils font des évaluations de la charge mentale.

Il distingue six dimensions de charge mentale : demande mentale, demande physique, demande temporelle, effort, performance et niveau de frustration. Les trois premières dimensions font référence aux demandes imposées au sujet (la demande mentale, physique et temporelle) et les autres trois font référence à l'interaction du sujet avec la tâche (l'effort, la frustration et le rendement ou la performance). La description de chaque dimension se trouve ci-dessous.

Demande mentale (faible/haute) Quel niveau de demande mentale et d'activité de perception étaient nécessaire (par exemple, penser, décider, calcul, mémoire, regarder, chercher, etc)? Les tâches étaient-elles faciles ou exigeantes, simples ou complexes?

Demande physique (faible/haute) Quel niveau d'activité physique était nécessaire (par exemple pousser, tirer, tourner, etc)? La tâche physique était-elle facile ou exigeante, lente ou rapide, reposante ou laborieuse?

Demande temporelle (faible/haute) Comment était le niveau de pression temporelle que vous avez ressenti en raison de la vitesse ou de l'allure à laquelle les éléments des tâches se sont produits?

Effort (faible/haut) Était-ce difficile (mentalement et physiquement) d'accomplir votre niveau de performance?

Performance (bonne/mauvaise) Comment pensez-vous avoir réussi dans l'accomplissement des objectifs de la tâche fixée par l'analyste (ou vous-même)? A quelle niveau êtes-vous satisfait de votre performance dans l'accomplissement de ces objectifs?

Niveau de frustration (faible/haut) Comment était votre niveau d'insécurité, de découragement, d'irritation, de stress et de contrariété lors de la tâche?

NASA-TLX est présenté au sujet après l'expérience. Chaque dimension reprend une échelle allant du pôle *faible* au pôle *haut*. Cette échelle est divisée en vingt points ce qui signifie que le sujet peut donner un avis nuancé à la dimension traitée.

Vu la diversité des dimensions et vu que certaines d'entre elles peuvent ne pas convenir pour l'expérience réalisée, l'examineur peut décider de mettre des ratio à chaque dimension. Par exemple, la dimension « demande physique » peut hériter d'un ratio faible voire nul. Ces poids sont fixés par l'expérimentateur. Il est ensuite possible de calculer la charge mentale à l'aide de la formule suivante :

$$\text{charge mentale} = DM * DMW + DP * DPW + DT * DTW + E * EW + P * PW + F * FW$$

où : DM, DP, DT, E, P, F sont les valeurs (données en pourcentage) fournies par l'utilisateur des différentes dimensions¹ et où DMW, DPW, DTW, EW, PW, FW les poids associés à chaque sous-échelle [HS88].

NASA-TLX fournit une technique simple, rapide et multi-dimensionnelle pour estimer la charge de travail d'un sujet. TLX est probablement la technique la plus utilisée pour estimer la charge de travail d'un sujet [Sta05].

1. respectivement : demande mentale, demande physique, demande temporelle, effort, performance et niveau de frustration.

L'expérimentation dans le génie logiciel

La révolution des technologies de l'information a entraîné, entre autres, la présence de logiciel dans de plus en plus de produits. Ce qui signifie qu'un grand nombre de logiciels ont été et seront encore développés. En 1968, le terme ingénierie du logiciel est inventé dans le but de mettre en place une discipline d'ingénierie qui cible le développement de systèmes logiciels. Ce chapitre a pour objectif de présenter la façon dont les études empiriques et surtout les expériences conviennent au contexte du génie logiciel. La section 6.1 définit le cadre des études empiriques dans le domaine du génie logiciel. Les différentes stratégies expérimentales qui peuvent être utilisées sont présentées en section 6.2. La section 6.3 clôture ce chapitre en précisant les étapes de conception et les activités d'une expérimentation.

Le contenu de ce chapitre est inspiré des ouvrages de Wholin et al. [WRH+00] et de Juristo et al. [JM10].

Sommaire

6.1	Génie logiciel empirique	46
6.1.1	Le contexte du génie logiciel	46
6.1.2	La science et le génie logiciel	46
6.2	Stratégies empiriques	47
6.2.1	Vue d'ensemble des stratégies expérimentales	47
6.2.2	Comparaison	48
6.3	Conception expérimentale	48
6.3.1	Processus expérimental	48
6.3.2	Définition de l'expérience	50
6.3.3	Planification de l'expérience	51
6.3.4	Exécution de l'expérience	55
6.3.5	Analyse et interprétation des données	56

6.1 Génie logiciel empirique

6.1.1 Le contexte du génie logiciel

Le génie logiciel est officiellement défini par l'association professionnelle *IEEE*¹ comme :

« l'application d'une approche systématique, disciplinée et quantifiable pour le développement, l'opération et la maintenance de logiciels. » [IEE90]

Trois aspects de la définition ont une importance pour le sujet que nous traitons [WRH+00] :

1. Ce domaine implique un processus logiciel à travers différentes phases de cycle de vie.
2. Il souligne le besoin d'une approche systématique et disciplinée.
3. Il met en évidence l'importance de la quantification.

L'utilisation d'étude empirique est liée à ces trois aspects.

Un modèle de processus logiciel est utilisé pour décrire les étapes et les activités à réaliser durant le développement d'un logiciel. Il est important de relever le fait que ce processus est crucial autant pour le développement d'une nouvelle application que pour la maintenance d'un produit existant.

Le développement d'un logiciel est une tâche ardue. En effet, les projets logiciels s'étalent sur de longues périodes et impliquent beaucoup de personnes. La première répercussion de ces caractéristiques se ressent directement sur le processus de développement logiciel qui en devient, lui aussi, plus complexe. Cette complexité du processus de développement engendre une difficulté à trouver et à optimiser le bon processus. Par ces observations, nous pouvons souligner l'importance d'une approche systématique et disciplinée durant la réalisation de ce processus logiciel. Pour l'améliorer, nous pouvons avoir recours à un processus d'amélioration de processus. Celui-ci inclut deux activités :

- Évaluation du processus logiciel.
- Évaluation d'une proposition d'amélioration du processus logiciel.

Malheureusement, une proposition d'amélioration est difficile à évaluer sans intervention humaine directe. L'utilisation d'une méthode impliquant des personnes est la seule évaluation réelle du processus ou d'une proposition d'amélioration de processus. C'est pour cette raison que les études empiriques sont cruciales. Elles permettent d'évaluer des processus et des activités humaines tout en garantissant une évaluation offrant de nombreux avantages. Plus concrètement, les expérimentations fournissent une méthode d'évaluation qui est systématique, disciplinée, quantifiable et contrôlée. Le secteur du génie logiciel ne fait pas exception à la règle. Ce qui implique que les chercheurs de ce domaine trouvent dans les études empiriques et, en particulier, dans les expériences, quantité de données qu'ils peuvent utiliser. Ainsi, les nouvelles inventions peuvent être évaluées avec celles qui existent déjà.

6.1.2 La science et le génie logiciel

Le génie logiciel est un sujet interdisciplinaire. Il s'étend des sujets techniques, comme les bases de données ou les systèmes d'exploitations, jusqu'aux sujets sociaux et psychologiques, en passant par des questions linguistiques. Cette discipline est basée sur la créativité et l'ingéniosité des personnes qui travaillent sur le terrain. Cependant, malgré ce caractère spontané, le génie logiciel n'en

1. Institute of Electrical and Electronics Engineers.

reste pas moins une science qui doit être considérée et traitée comme telle [WRH+00]. Ce qui implique l'utilisation de méthodes scientifiques. Pour pouvoir réaliser des recherches scientifiques en génie logiciel, nous devons comprendre les méthodes qui s'offrent à nous, leurs limitations et à quel moment elles peuvent être utilisées.

Dans [Glass94], quatre méthodes de recherche se rapportant au domaine du génie logiciel ont été résumées :

- **la méthode scientifique** : le monde est observé et un modèle est construit sur base des observations. Traditionnellement cette méthode est utilisée dans les domaines appliqués (exemple : évaluer les performances d'un réseau de télécommunication en le simulant).
- **la méthode de l'ingénierie** : la solution courante est étudiée et, ensuite, des changements sont proposés puis évalués. L'industrie est le principal utilisateur de cette méthode.
- **la méthode empirique** : un modèle est proposé et évalué à travers des études empiriques (par exemple des études de cas ou des expériences). Cette méthode était traditionnellement utilisée dans les sciences sociales et la psychologie, disciplines pour lesquelles il est impossible de se baser sur la nature, comme en physique. Le génie logiciel est gouverné par le comportement humain au travers des personnes qui développent les logiciels. C'est pourquoi nous ne devons pas nous attendre à trouver des règles formelles ou des lois en génie logiciel.
- **la méthode analytique** : une théorie formelle est proposée et comparée avec des observations empiriques. Habituellement, cette méthode est utilisée dans des domaines très formels, comme le génie électrique ou l'informatique.

Il convient de noter que nous ne qualifions pas les méthodes analytiques, scientifiques et la méthode de l'ingénierie d'inappropriées pour le génie logiciel. Elles peuvent être nécessaire pour certaines parties du domaine. La volonté de comprendre et d'identifier les relations existantes entre différentes variables constitue la principale raison d'utilisation de l'expérimentation dans le génie logiciel.

6.2 Stratégies empiriques

6.2.1 Vue d'ensemble des stratégies expérimentales

Il existe deux approches différentes se rapportant à deux types de paradigmes empiriques différents [WRH+00] :

- **Recherche qualitative** : elle concerne l'étude d'objets dans leur environnement naturel. Un chercheur qualitatif essaie d'interpréter un phénomène basé sur les explications fournies par des personnes. Ce type de recherche s'intéresse à la découverte de causes relevées par les sujets de l'étude et leur compréhension du problème à résoudre. Les sujets sont les personnes qui prennent part à l'expérience afin de pouvoir évaluer un objet.
- **Recherche quantitative** : elle concerne la quantification d'une relation ou la comparaison de deux ou plusieurs groupes. Le but est d'identifier une relation de cause à effet. Il s'agit souvent de la mise en place d'une expérience contrôlée ou d'une récolte de donnée au travers d'une étude de cas. Les investigations quantitatives sont appropriées lorsque l'on teste l'effet de certaines manipulations ou certaines activités.

Wholin *et al.* [WRH+00] distinguent 3 types de stratégies empiriques : *l'enquête*, *l'étude de cas* et *l'expérience*. Le choix pour l'une de ces 3 stratégies doit être fait en fonction du but de l'évaluation, de la portée et des conditions de l'étude empirique.

Enquête : Une enquête est souvent une investigation réalisée rétrospectivement, par exemple lors-

qu'un outil ou une technique est déjà utilisée depuis un certain laps de temps. Des interviews ou des questionnaires permettent de récolter les données qualitatives ou quantitatives. Des conclusions descriptives et explicatives sont dérivées de l'analyse des résultats.

Cas d'étude : Les études de cas sont utilisées pour suivre des projets, des activités ou des affectations. Les données sont récoltées dans un but spécifique durant toute la durée de l'étude. En fonction de celles-ci, des analyses statistiques peuvent être réalisées. L'étude de cas est généralement mise en place pour suivre un attribut spécifique ou pour établir des relations entre différents attributs.

Expérience : Les expériences sont traditionnellement accomplies dans un laboratoire, afin de garantir un haut niveau de contrôle. L'objectif de l'expérience est de manipuler une ou plusieurs variables et d'assigner les autres variables à une valeur fixée. Les effets de la manipulation sont mesurés, et des études statistiques sont réalisées sur ceux-ci.

6.2.2 Comparaison

Les pré-requis concernés par une étude limitent souvent le choix de la stratégie empirique. Une comparaison des stratégies peut se faire suivant un nombre de facteurs différents. Wholin *et al.* [WRH+00] prennent en compte les quatre facteurs suivants lors de la comparaison des stratégies : la valeur de contrôle que le chercheur possède sur l'étude, le degré de liberté que le chercheur a sur les mesures à collecter, le coût de l'étude et la facilité avec laquelle une réplication de l'étude peut être réalisée. Leur tableau de comparaison est repris dans le tableau 6.1.

Facteur	Enquête	Cas d'étude	Expérience
Contrôle d'exécution	Non	Non	Oui
Contrôle des mesures	Non	Oui	Oui
Coût d'investigation	Bas	Moyen	Haut
Facilité de réplication	Haut	Bas	Haut
Type de recherche	Qualitative & Quantitative	Qualitative & Quantitative	Quantitative

Tableau 6.1 – comparaison des stratégies empiriques [WRH+00]

6.3 Conception expérimentale

6.3.1 Processus expérimental

Le point de départ de toute expérience réside dans une idée de relation entre une cause et un effet qu'il est possible de formuler sous forme d'une hypothèse. L'expérimentation est utilisée pour évaluer une conviction de cause à effet. Lors de la conception d'une expérience, le chercheur a toujours un certain nombre de traitements (valeurs que les variables étudiées peuvent prendre) sur lesquels il a le contrôle. Lorsque l'expérience est effectuée, il est possible d'observer les relations entre les traitements et les résultats. Le principe de base d'une expérience est illustré par la figure 6.1. Le processus présenté dans cette section assure que toutes les actions appropriées sont prises, ce qui favorise la réussite de l'expérience. L'objectif du processus est de fournir un support pour la mise en place et la réalisation d'une expérimentation.

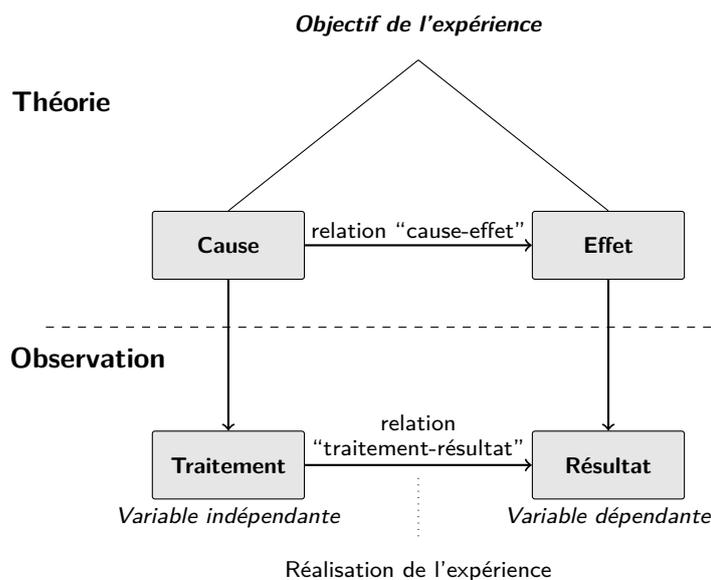


FIGURE 6.1 – Le principe expérimental [WRH+00]

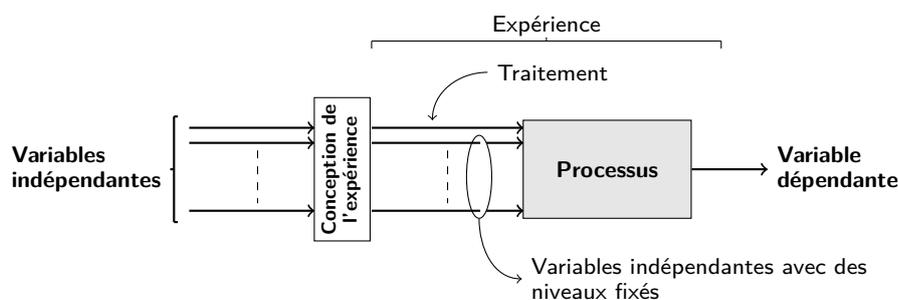


FIGURE 6.2 – Illustration d'une expérience [WRH+00]

Comme nous l'avons vu plus haut, lors de la production d'une expérience, le chercheur manipule une ou plusieurs variables qui peuvent avoir une valeur attribuée. En réalité, il existe deux sortes de variables dans une expérience : les *variables indépendantes* et les *variables dépendantes*. Les **variables indépendantes** correspondent aux variables qui sont manipulées et contrôlées dans le processus. Les **variables dépendantes** (aussi appelés *variables de réponse*) sont les variables sur lesquelles nous voulons étudier l'effet du changement des variables indépendantes. Une expérimentation étudie l'effet produit par le changement d'une ou plusieurs variables indépendantes. Ces variables sont également appelées *facteurs*. Les autres variables indépendantes sont fixées à une certaine valeur durant l'expérimentation. Ceci permet d'analyser si le facteur ou la variable a un effet particulier. Un *traitement* est une valeur particulière d'un facteur. La figure 6.2 illustre les différents éléments que nous venons de décrire.

Les traitements vont être attribués à une combinaison d'*objets* et de *sujets*. Un document qui doit être examiné avec différentes techniques d'inspection est un exemple d'objet. Les sujets² sont les personnes qui appliquent les traitements. La plupart du temps, les variables indépendantes contiennent certaines caractéristiques des objets et des sujets. Une expérience consiste en un ensemble de *tests*. Un test est une combinaison d'un traitement, d'un sujet et d'un objet.

2. Parfois le terme *participant* est utilisé.

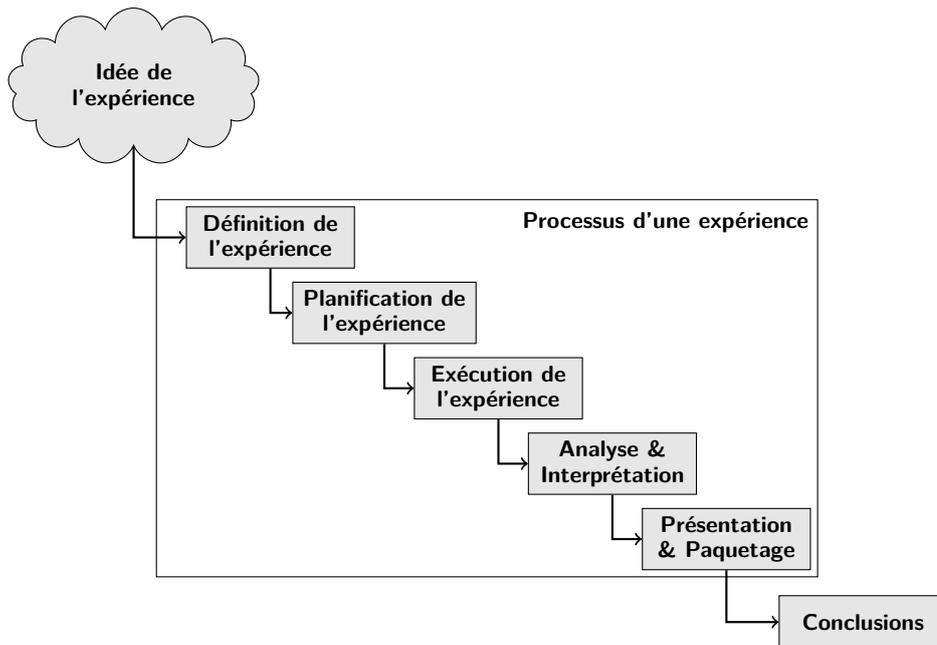


FIGURE 6.3 – Activités du processus d'expérimentation [WRH+00]

Comme illustré sur la figure 6.3, le processus d'expérimentation est divisé en différentes activités :

1. Définition de l'expérience.
2. Planification de l'expérience.
3. Exécution de l'expérience.
4. Analyse et interprétation des données.
5. Présentation et paquetage.

Les sous-sections 6.3.2, 6.3.3, 6.3.4 et 6.3.5 présentent le contenu de ces différentes étapes. L'activité de présentation et paquetage présente les aspects à prendre en compte lors de la publication d'un article ou d'un rapport. Cette activité ne sera pas décrite dans ce document car elle n'est pas essentielle dans le cas qui nous concerne.

6.3.2 Définition de l'expérience

Dans la phase de définition, les fondations de l'expérience sont établies. Des mauvaises fondations peuvent entraîner une impossibilité d'utilisation de l'expérimentation pour étudier ce qui était prévu à la base. Le but de l'activité de définition est de préciser les objectifs de l'expérience selon un cadre défini. Wholin *et al.* [WRH+00] se basent sur le paradigme GQM (« goal, question, metric » ou approche par les buts) pour assurer une définition des aspects importants d'une expérimentation avant la planification et la réalisation de celle-ci. Ces auteurs préconisent d'établir la définition d'une expérience de la manière suivante :

“Analyse du(des) <Objet(s) d'étude>
 dans le but de <Objectif>
 selon la <Focalisation sur la qualité>
 sous l'angle de <Perspective>
 dans le contexte du <Contexte> ”

[WRH+00]

Objet d'étude : L'objet d'étude représente l'entité étudiée dans l'expérience. Il peut s'agir d'un produit, d'un processus, d'une ressource, d'un modèle, d'une métrique ou d'une théorie.

Objectif : L'objectif désigne l'intention de l'expérimentation. L'évaluation de l'impact de deux techniques différentes ou la caractérisation de la courbe d'apprentissage d'une organisation sont des exemples d'objectif.

Focalisation sur la qualité : Cet élément représente l'effet primaire étudié dans l'expérience. Des exemples de focalisation de qualité sont l'efficacité, le coût, etc.

Perspective : La perspective détermine le point de vue à partir duquel les résultats de l'expérience seront interprétés.

Contexte : Le contexte est l'environnement dans lequel l'expérience est réalisée. Le contexte comprend la détermination des sujets et des objets utilisés au sein de l'expérience.

6.3.3 Planification de l'expérience

Après avoir posé le *pourquoi* dans la définition de l'expérience, le planning va déterminer *comment* l'expérience va être menée. Cette étape doit être accomplie correctement afin de ne pas produire des résultats altérés ou détruits. Comme illustré sur la figure 6.4, sept sous-activités sont nécessaires pour la planification d'une expérience :

1. La sélection du contexte.
2. La formulation des hypothèses.
3. La sélection des variables.
4. La sélection des sujets.
5. La conception de l'expérience.
6. L'instrumentation.
7. L'évaluation de la validité.

Sélection du contexte

Dans le meilleur des cas, une expérience devrait être réalisée sur des projets réels avec des sujets professionnels. Malheureusement, mener ce genre d'expérience est coûteux et peut comporter des risques de délai non négligeables dans l'évolution d'un produit logiciel réel. Réaliser l'expérience sur des projets étudiants peut être une alternative intéressante. Cela conduit à des projets plus économiques et plus facile à contrôler, mais qui sont liés à un certain contexte. Ce compromis implique un choix entre la réalisation d'une étude valide pour un certain contexte ou valide pour le domaine du génie logiciel en général. Ainsi, l'environnement d'une expérimentation peut être caractérisé suivant quatre dimensions :

- Hors-ligne vs. en-ligne.
- Étudiant vs. professionnel.
- Jouet vs. monde réel.
- Spécifique vs. général.

Formulation des hypothèses

Le test d'hypothèse constitue la base de l'analyse statistique d'une expérience. Des hypothèses sont établies formellement et les données collectées durant l'expérimentation sont utilisées pour réfuter celles-ci. Dès qu'une hypothèse est rejetée, une conclusion peut être tirée. Deux types d'hypothèses doivent être formulées durant la phase de planification :

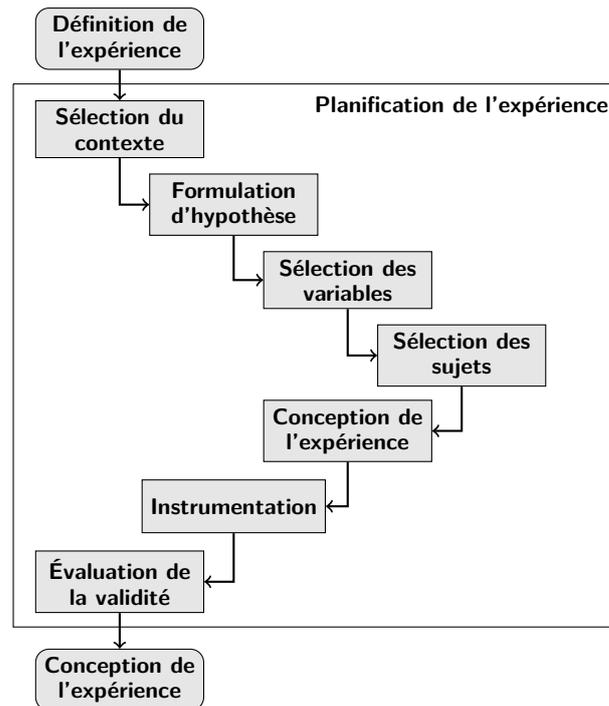


FIGURE 6.4 – Phase de planification d’une expérience [WRH+00]

- L’hypothèse nulle, H_0 : cette hypothèse indique qu’il n’existe pas de relation de cause à effet dans l’expérience, les différences présentes dans les observations sont le fruit du hasard. En générale, l’expérimentateur cherche à rejeter cette hypothèse nulle.
- L’hypothèse alternative, $H_a, H_1, \text{etc.}$: elle est l’hypothèse en faveur de laquelle l’hypothèse nulle est rejetée.

Trois types de risques émanent du test d’hypothèses :

- L’erreur de type I : une erreur de type I se produit lorsqu’un test statistique indique une relation alors qu’il n’en existe en réalité aucune. L’erreur de type I peut s’exprimer au travers de la probabilité suivante :

$$P(\text{erreur de type I}) = P(\text{rejeter } H_0 | H_0 \text{ vrai})$$

- L’erreur de type II : une erreur de type II se produit lorsqu’un test statistique n’indique pas de relation alors qu’il en existe une en réalité. L’erreur de type II peut s’exprimer au travers de la probabilité suivante :

$$P(\text{erreur de type II}) = P(\text{ne pas rejeter } H_0 | H_0 \text{ faux})$$

- La puissance : la puissance d’un test statistique est représentée par la probabilité qu’un test révèle une vraie relation si H_0 est faux. La puissance peut s’exprimer au travers de la probabilité suivante :

$$P(\text{Puissance}) = P(\text{rejeter } H_0 | H_0 \text{ faux}) = 1 - P(\text{erreur de type II})$$

Sélection des variables

Avant de pouvoir concevoir l'expérience, il est nécessaire de choisir les variables indépendantes et les variables dépendantes :

- Les variables indépendantes sont celles dont nous avons le contrôle et qui changent dans l'expérience. Choisir les bonnes variables n'est pas facile et nécessite souvent des connaissances du domaine étudié dans l'expérience. Ces variables doivent avoir certains effets et doivent pouvoir être contrôlables.
- Les effets des traitements sont mesurés grâce aux variables dépendantes. Dès que celles-ci ont été choisies, les hypothèses peuvent être affinées.

Sélection des sujets

La sélection des sujets possède une place importante lorsque l'on mène une expérience. Cette sélection est étroitement liée à la généralisation des résultats de l'expérimentation. Pour pouvoir généraliser les résultats à une population désirée, la sélection doit être représentative de cette population. Sa taille influence les résultats, plus l'échantillon est grand, plus les erreurs lors de la généralisation deviennent minimales.

Conception de l'expérience

Une expérience consiste en une série de tests de traitement. Les séries de tests doivent être soigneusement planifiées et conçues afin d'obtenir le maximum de l'expérience. Lors d'une étape de conception, l'expérimentateur décrit comment les tests sont organisés et réalisés. La conception et les analyses statistiques sont étroitement liées, de telle sorte que le choix d'une certaine conception affecte les analyses et inversement. Nous devons donc consulter les hypothèses pour voir quels tests statistiques sont nécessaires au rejet de l'hypothèse nulle. Durant la conception, l'expérimentateur décide le nombre de tests que l'expérimentation doit avoir pour assurer la visibilité des effets des traitements.

Lors de la conception d'une expérimentation, certains principes généraux sont à prendre en compte :

1. La *randomisation* est l'un des principes de conception le plus important. Elle est utilisée pour favoriser l'utilisation d'observations provenant de variables aléatoires indépendantes. Elle s'applique sur la répartition des objets et des sujets ainsi que sur l'ordre dans lesquels les tests sont menés. La randomisation est utilisée pour atténuer l'effet d'un facteur présent qui ne fait pas partie des traitements. Son objectif est de diminuer les effets dus à des aspects systématiques de l'expérience.
2. Parfois un facteur possède un effet sur les résultats qui est sans intérêt pour l'expérience mise en place. Pour contrer cette menace, il existe un principe général dénommé *blocage*. Le blocage est utilisé pour éliminer systématiquement l'effet indésirable dans la comparaison entre les traitements. À l'intérieur d'un bloc, l'effet non désiré est le même et nous pouvons étudier l'effet du traitement sur ce bloc. Cependant, ce principe peut être utilisé uniquement si les effets du facteur qui pose problème sont connus et contrôlables.
3. Une conception *équilibrée* est une conception pour laquelle chaque traitement est assigné au même nombre de sujet. Bien que ce principe n'est pas indispensable, il est néanmoins désirable car il simplifie et renforce l'analyse statistique des données.

Wholin *et al.* énoncent les quatre types de conception les plus utilisés dans les études empiriques :

- **Un facteur avec deux traitements** Ce type de conception permet de comparer les deux traitements l'un avec l'autre. Dans ce type de conception, la méthode la plus répandue est la comparaison des moyennes des variables dépendantes pour chaque traitement.
- **Un facteur avec plus de deux traitements** Comme le type précédent, l'expérimentateur veut comparer les traitements entre eux. Cette comparaison est également effectuée sur la moyenne des traitements.
- **Deux facteurs** Une expérience à deux facteurs est plus complexe à concevoir. Dans ce type de conception, trois hypothèses sont nécessaires : une pour l'effet du premier facteur, une seconde pour l'effet du second facteur et une troisième pour l'interaction de ces deux facteurs. La conception peut prendre des formes différentes si les facteurs sont imbriqués (conception de type *imbriquée à deux étages*) ou non (conception de type *factorielle 2 * 2*).
- **Plus de deux facteurs** Beaucoup d'expériences impliquent plus de deux facteurs. Ce type de conception est appelée la *conception factorielle*. Dans cette conception, l'effet des variables dépendantes peut être influencé par chaque facteur séparément, mais également par chaque interaction de facteur.

Instrumentation

Les objets, les lignes de conduites et les instruments de mesures font partie des outils de l'expérience. Ces instruments sont choisis lors de la planification de l'expérience.

Les objets de l'expérience couvrent un large domaine. Une spécification ou des documents de code peuvent être utilisés comme tels. Lors de la planification d'une expérimentation, il est très important de choisir les objets les plus appropriés. Les lignes de conduites permettent de guider le sujet durant l'expérience. Elles contiennent par exemple la description du processus et une liste de contrôle. Les lignes de conduites peuvent également comporter des tutoriels se rapportant à l'utilisation des méthodes employées.

Les mesures d'une expérience sont réalisées via la collection des données. Dans la plupart des expériences, ces mesures sont récoltées manuellement à partir d'entrevues. Préparer la forme et les questions de ces entrevues fait partie de la sous-activité d'instrumentation de l'expérience. En conclusion, le but général de l'instrumentation consiste à fournir des moyens nécessaires à la réalisation et à la surveillance de l'expérience, sans affecter le contrôle de celle-ci. Si cette dernière condition est violée, alors on considère que les résultats de l'expérience sont invalides.

Évaluation de la validité

Il est primordial de considérer la question de la validité des résultats dès la phase de planification. En effet, une telle démarche favorise une validation adéquate des résultats de l'expérience. Pour détecter si la validité est réellement adéquate, on se réfère aux résultats qui doivent être valables pour la population directement concernée par l'expérience. Premièrement, les résultats doivent être valides pour la population dont est issue l'échantillon. Ensuite, la généralisation des résultats à une population plus large est importante.

La littérature comprend plusieurs schémas de classification de menaces à la validité. Nous nous contenterons du modèle présenté par Wholin *et al.* et inventé par Cook et Campbell. Cette catégorisation comporte les quatre types de menaces suivantes :

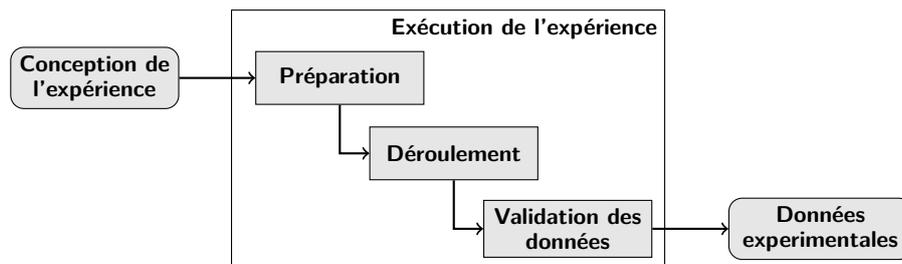


FIGURE 6.5 – Phase d'exécution d'une expérience [WRH+00]

- **La validité de conclusion** : Elle est concernée par la relation entre le traitement et les résultats. Il faut s'assurer qu'il y a une relation statistique significative.
- **La validité interne** : La causalité de la relation entre le traitement et les résultats fait partie de la validité interne. Seul le traitement doit avoir produit le résultat.
- **La validité de construction** : Cette validité concerne la relation entre la théorie et l'observation. Si la relation entre l'effet et la cause est bien causale, il faut s'assurer que le traitement reflète correctement la construction de la cause et que le résultat reflète correctement la construction de l'effet.
- **La validité externe** : La généralisation des résultats concerne ce type de menace. Les résultats peuvent-ils être généralisés en dehors du cadre de celle-ci s'il existe bien une relation causale entre la construction de la cause et l'effet ?

6.3.4 Exécution de l'expérience

L'expérience peut être menée dès que sa conception et sa planification sont finies. Durant la phase d'exécution, l'expérimentateur rencontre les sujets, et les traitements leur sont appliqués. Trois étapes sont nécessaires à cette phase. La préparation est la première étape. Durant celle-ci, les sujets sont choisis et les formulaires sont préparés. La deuxième étape, l'exécution proprement dite est l'étape où les sujets réalisent leurs tâches et les données sont collectées. La dernière étape, la validation des données, consiste à valider les données collectées. Ces différentes étapes sont illustrées sur la figure 6.5 et sont détaillées ci-dessous.

Préparation

L'exécution est facilitée par une bonne préparation. Cette préparation passe par la sélection et l'information des participants et par la préparation du matériel de l'expérience.

Il est essentiel que les sujets soient motivés et prêts à participer à l'ensemble de l'expérimentation. Certains aspects doivent être pris en compte lorsque des personnes participent à une expérience :

- **L'obtention du consentement** : Les participants doivent comprendre et être d'accord avec les objectifs de la recherche. Une mauvaise compréhension peut influencer négativement les résultats de l'expérience. Il est également important de décrire comment les résultats de l'expérimentation vont être utilisés et publiés. L'expérimentateur doit pouvoir laisser la possibilité au sujet de se retirer de l'expérience à tout moment.
- **La confidentialité des résultats** : Il arrive que les résultats obtenus lors de l'expérience soient sensibles pour les participants. Dans ce cas, il est primordial d'assurer une confidentialité des résultats.
- **L'incitation** : Utiliser un certain avantage peut aider à attirer les sujets à participer à l'expérimentation. Sa valeur ne doit cependant pas être trop élevée pour éviter de solliciter des sujets

non sérieux.

- **La tromperie** : Tromper ou trahir les participants n'est pas préférable. Si des méthodes alternatives sont disponibles, elles doivent être privilégiées à la tromperie.

Comme nous l'avons énoncé précédemment, tous les instruments de l'expérimentation doivent être prêts avant l'exécution de celle-ci. Ils contiennent les objets, les lignes de conduites et les formes de mesures de l'expérience.

Exécution

L'expérimentation peut être réalisée de différentes façons. Certaines d'entre elles permettent de faire participer tous les sujets en une seule fois, alors que d'autres ne peuvent être achevées que par un seul sujet à la fois.

Durant l'expérimentation, la collection de données peut être faite manuellement (au travers de formulaires ou d'entrevues) ou automatiquement. L'avantage des formulaires est qu'ils ne demandent pas beaucoup d'effort pour l'expérimentateur. Mais le fait de ne pas pouvoir relever directement les inconsistances, incertitudes et les défauts du formulaire constitue un inconvénient. Ce type de manquement ne peut être facilement détecté avant la collection des données sauf si le participant attire l'attention sur les fautes ou pose des questions. La technique de l'entrevue offre la possibilité de mieux communiquer avec le sujet pendant la collection de données. Malheureusement, celle-ci demande plus d'effort de la part du chercheur.

Validation

Une fois les données collectées, le chercheur doit être en mesure de vérifier si la collection a été faite d'une façon correcte. Une mauvaise compréhension du formulaire peut entraîner des données invalidées. Une autre source d'erreur peut provenir de l'accomplissement de l'expérience par un sujet non sérieux. Dans ce cas, ces données devront être écartées des résultats.

6.3.5 Analyse et interprétation des données

Après avoir collecté les données expérimentales durant la phase d'opération, l'expérimentateur doit pouvoir tirer des conclusions basées sur ces données. L'interprétation quantitative peut être effectuée en trois étapes : les *statistiques descriptives* qui visualisent les tendances centrales, la *réduction de l'ensemble des données* qui exclue les données invalides et fausses et, enfin, les *tests d'hypothèses* qui évaluent statistiquement les hypothèses.

Statistiques descriptives

Les statistiques descriptives se composent de représentations graphiques et numériques d'ensemble de données. Ce type de statistiques est utilisé pour présenter des aspects intéressants d'une collection de données. Leur but est d'obtenir une impression sur la façon dont les données sont distribuées. Cette première étape sera réalisée avant les tests d'hypothèses, afin de mieux comprendre la nature des données et d'identifier les données fausses ou anormales. La *boîte à moustaches* est une représentation graphique d'éléments de statistiques descriptives. Cette méthode met en évidence des mesures de tendances centrales telles que les *quartiles* et la *médiane*.

Médiane La médiane représente la valeur centrale d'un échantillon trié par ordre croissant (ou décroissant). Si l'échantillon comporte un nombre pair de valeur (n), alors la médiane est égale à la

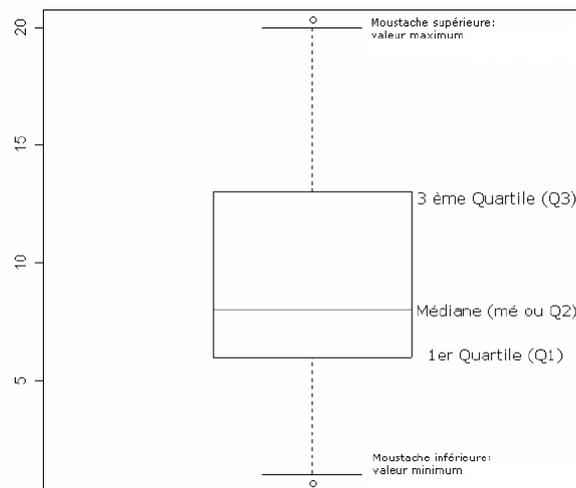


FIGURE 6.6 – Exemple de boîte à moustaches

moyenne entre le $\frac{n}{2}$ -ième élément et le $\frac{n+1}{2}$ -ième élément de l'échantillon. Dans le cas contraire, la médiane est égale à la valeur du $\frac{n+1}{2}$ -ième élément de l'échantillon.

Quartiles Les quartiles ont pour objectif de diviser un échantillon trié en quatre parts égales, de sorte que chaque partie représente un quart de l'échantillon de la population. Nous distinguons trois quartiles : le 1^{er} quartile (Q_1), le 2^e quartile (Q_2) et le 3^e quartile (Q_3). Les 25% de valeurs inférieures sont séparées par Q_1 , c'est-à-dire que 75% des données de l'échantillon lui sont supérieures. Q_2 correspond à la médiane. Q_3 sépare l'échantillon afin que 25% des données lui soient supérieures. Pour un échantillon de taille n , Q_1 est égale au $\frac{n}{4}$ -ième élément et Q_3 est égale au $\frac{n*3}{4}$ -ième élément de l'échantillon. L'ensemble des valeurs comprises entre Q_1 et Q_3 est nommé l'écart *interquartile*, il rassemble 50% des effectifs.

Boîte à moustaches La boîte à moustaches est une méthode utilisée pour afficher la distribution d'un ensemble de données continues. Ce type de graphique comprends 5 éléments : la médiane, le premier et le troisième quartile (et donc l'écart interquartile), la valeur minimum et la valeur maximum de l'ensemble de données. La figure 6.6 est un exemple de boîte à moustache.

Réduction des données

Le résultat des tests statistiques dépend essentiellement de la qualité des données qui sont fournies en entrée. Les conclusions tirées pourraient être incorrectes si les données ne représentent pas ce que nous voulons qu'elles représentent. Les erreurs dans une collection de données peuvent provenir d'erreurs systématiques ou apparaître sous forme de valeurs extrêmes³. Une valeur extrême est une donnée qui est plus grande ou plus petite qu'elle ne devrait être au regard des autres données de l'ensemble. Plusieurs moyens existent pour identifier ces valeurs atypiques ; l'une des plus efficace est le diagramme de dispersion. Les valeurs extrêmes sont également visibles dans les boîtes à moustaches : le caractère « o » présent au dessus de la moustache supérieure et en dessous de la moustache inférieure de la figure 6.6 en représente deux.

La réduction des données est en relation avec la validation des données discutée dans la sous-section précédente (sous-section 6.3.4). La validation des données se charge de l'identification des données fausses en se basant sur l'exécution de l'expérience. D'un autre coté, le type de réduction de

3. Aussi nommées *marginales*, *atypiques* ou *exceptionnelles*.

données, discuté au sein de cette section, identifie les données atypiques en observant les données collectées et en tenant compte des statistiques descriptives.

Une fois les données marginales identifiées, il faut pouvoir décider ce que l'on peut en faire. Cette décision sera prise en fonction de l'analyse des causes de cette position extrême dans le diagramme de dispersion. Si elle est due à un évènement étrange ou rare, elle pourra être exclue des résultats. S'il s'agit d'un évènement qui pourrait encore se produire, il est déconseillé de l'exclure des analyses. Si les données marginales ont pour cause une variable non considérée dans l'expérience, il est important de baser les calculs et les modèles sur celle-ci.

Tests d'hypothèses

Comme il l'a été précédemment énoncé dans la section 6.3.3, le test d'hypothèse a pour but de vérifier s'il est possible de rejeter l'hypothèse nulle, H_0 , basée sur échantillon d'une distribution statistique. Dans la plupart des cas, un seul paramètre influence la distribution. Formuler la distribution et assigner une valeur au paramètre constitue la mise en place de H_0 . Le paramètre choisi dans l'hypothèse sera alors testé.

Pour tester H_0 , une unité de test t est définie et une zone critique C est donnée. C est une partie de la zone autour de laquelle t varie. Suivant ces formulations, le test significatif peut être formulé comme :

- Si $t \in C$, H_0 est rejeté.
- Si $t \notin C$, H_0 n'est pas rejeté.

L'hypothèse nulle doit être formulée négativement. Si elle ne peut être rejetée, rien ne peut être affirmé sur les résultats. Par contre si l'hypothèse est rejetée, il peut être déclaré que H_0 est fautive avec un niveau significatif déterminé (α). Lorsqu'un test d'hypothèse est effectué, il est souvent possible de calculer la *p-value*. Cette valeur exprime la probabilité d'obtenir des résultats au moins aussi extrêmes que ceux observés, en supposant que l'hypothèse nulle est vraie. Si la *p-value* est strictement inférieur à α , alors il est possible de rejeter l'hypothèse nulle.

Les tests d'hypothèses sont classés en deux catégories :

1. **Les tests paramétriques** Ils sont fondés sur un modèle qui comporte une distribution spécifique. Le plus souvent, certains des paramètres impliqués dans les tests paramétriques sont normalement distribués. Si les paramètres ne peuvent être mesurés sur une échelle d'intervalles, ce type de test ne pourra être utilisé.
2. **Les tests non-paramétriques** Ils sont plus généraux que les tests paramétriques. Ils sont basés sur des présuppositions moins fortes et donc sont utilisés lorsqu'ils ne respectent pas les conditions des tests paramétriques.

Deux facteurs sont à prendre en compte lors du choix d'un type de test :

- **L'applicabilité** Il est important que les suppositions concernant les échelles et concernant la distributions des paramètres soient réalistes.
- **La puissance** La puissance des tests paramétriques est plus élevée que celle des non-paramétriques.

Il existe un nombre important de tests paramétriques et non paramétriques. Comme nous pouvons l'observer dans le tableau 6.2, ceux-ci peuvent être triés suivant le type de conception de l'ex-

Conception	Paramétrique	Non-paramétrique
Un facteur, un traitement	/	Chi-2 & Binomial test
Un facteur, deux traitements	t-test & F-test	Mann-Whitney & Chi-2
Un facteur, plus de deux traitements	ANOVA	Kruskal-Wallis & Chi-2
Plus d'un facteur	ANOVA	/

Tableau 6.2 – Résumé des tests d'hypothèses paramétriques/non-paramétriques pour les différents type de conception

périence et suivant leur caractère paramétrique/non-paramétrique. Ces différents tests sont résumés dans l'ouvrage de Wholin *et al.* [WRH+00, p. 99-107]. Le lecteur désirant plus d'informations pourra lire [JM10], [SC88] ou [MS88]. Par ailleurs, il est important de préciser que les tests à plus de deux traitements identifient la présence de différences entre les échantillons, mais ne déterminent pas où se trouve précisément ces différences. Il est alors nécessaire d'utiliser des tests tel que le t-test, le F-test, le test Mann-Whitney ou le test Chi-2. En effet, appliqués sur l'ensemble des paires de traitement possible, ces tests révèlent quelles sont les paires qui possèdent réellement une différence.

Matériel

La réalisation d'une expérience oculométrique passe par la collecte de données et l'analyse de celles-ci. Ces deux étapes nécessitent l'utilisation de différents outils matériels ou logiciels. Ce chapitre est dédié à la présentation des outils que nous avons utilisés lors de notre expérience. La section 7.1 s'intéresse principalement aux matériels physiques mis à notre disposition, c'est-à-dire aux oculomètres. Cette section contient principalement une explication de leur fonctionnement et une présentation des logiciels dont ils dépendent. La section 7.2 présente le logiciel développé par le laboratoire Ptidej, prénommé TAUPE, et utilisé pour traiter les données collectées. La structure ainsi que les fonctionnalités y seront détaillées. Une attention plus particulière sera portée sur les améliorations que nous avons réalisées.

Sommaire

7.1	Oculomètre	62
7.1.1	Eye-LinkII	62
7.1.2	FaceLAB	62
7.2	TAUPE	63
7.2.1	TAUPE v2.0	64
7.2.2	TAUPE v2.1	67

7.1 Oculomètre

Un oculomètre (ou eye-tracker) est un appareil qui analyse des images de l'œil humain capturées par une caméra, souvent en lumière infrarouge, pour calculer la direction du regard du sujet. En fonction de la précision souhaitée, différentes caractéristiques de l'œil sont analysées.

Le matériel mis à notre disposition à l'école Polytechnique de Montréal nous permettait de travailler avec deux oculomètres : *Eye-linkII* et *FaceLAB*.

Le premier est un système intrusif, c'est-à-dire que le sujet doit porter du matériel. Ce système est composé de trois caméras miniatures montées sur un casque. Cependant, ce système souffre d'une précision qui n'est pas toujours satisfaisante. De plus, le sujet se doit de rester assez immobile afin de permettre à cet appareil de pouvoir enregistrer les données correctement.

Le deuxième oculomètre est un système plus récent et non-intrusif, ce qui signifie que le sujet ne porte rien. Ce système est composé d'un ordinateur (portable ou fixe) auquel sont reliées deux caméras infrarouges.

7.1.1 Eye-LinkII

Le système Eye-link II se compose de trois caméras miniatures montées sur un casque. Plus précisément, il s'agit de deux caméras oculaires qui permettent le suivi du regard binoculaire ou une sélection facile de l'œil dominant du sujet sans la reconfiguration mécanique exigée par la plupart des eye-trackers par casque. La troisième caméra permet un suivi plus précis du point de vision du sujet.

La suite logicielle associée à cet appareil est capable de générer des fichiers de type .edf, format propriétaire qu'il est néanmoins possible de convertir en XML (via un programme nommé edf2xml). Ce fichier comporte une suite d'éléments XML. Chaque élément de ce fichier comporte un temps de départ, un temps de fin (en millisecondes) et une durée (en millisecondes). Certains éléments sont propres aux fixations, d'autres représentent des saccades. Dans le premier cas, l'élément contient les coordonnées (x,y). Le second type d'élément possède des coordonnées de départ (x1,y1) et d'arrivée (x2,y2), ainsi que des informations concernant son amplitude et sa vitesse maximale.

7.1.2 FaceLAB

Cet oculomètre est caractérisé par deux caméras, comme nous pouvons le voir sur la figure 7.1. Ces deux caméras servent à enregistrer la position de la tête. Pour ce faire, le système se sert des différents éléments du visage (les sourcils, les lèvres, le nez, etc) pour aider les caméras à détecter les yeux. Une fois les éléments faciaux enregistrés par les caméras, un test de calibration, qui consiste à regarder neuf points, positionnés pour occuper tout l'écran, est effectué afin de corriger automatiquement les offsets du sujet. Une fois que l'utilisateur a confirmé les positions de ces éléments et le test, FaceLAB est capable de suivre et d'adapter *automatiquement* les coordonnées du regard du sujet. En effet, avec cette configuration, il est possible de suivre le regard d'un sujet même si celui-ci tourne la tête.

La configuration de FaceLAB a été facilitée par la documentation présente. Les différentes vidéos de Lempereur et De Smet¹ sont d'une aide non négligeable. Cette configuration dépend de deux facteurs. Tout d'abord, il faut bien paramétrer le visage du sujet.

1. <http://www.ptidej.net/research/taupe/videos/>

Pour chaque sujet, il est important de reconfigurer les différents éléments du visage (deux sujets n'ont pas la même physiologie du visage) et d'effectuer une calibration de cette nouvelle configuration. De plus, il faut configurer le « world model » qui prend en paramètre les emplacements de l'écran, des caméras et du sujet. Normalement, si personne ne touche aux caméras, cette configuration ne nécessite pas d'être refaite à chaque fois.

Un oculomètre fonctionne toujours avec un logiciel qui permet de gérer les informations qu'il récolte. Pour FaceLAB, il fonctionne de pair avec le logiciel *GazeTracker*TM (de EyeResponse). Ce logiciel permet d'afficher des données, le code source et les instructions pour créer un plan d'expérimentation. Dans ce logiciel, une expérimentation est composée d'une suite d'images, qui sont dévoilées au sujet l'une après l'autre, à la manière d'une présentation powerpoint. Pour chaque image, le parcours visuel produit par le sujet sera enregistré. Cependant, ce logiciel n'est pas libre, ni à code ouvert et pas gratuit. Ce qui implique qu'aucune modification, même intéressante ne peut être ajoutée à ce logiciel. Le logiciel TAUPE nous permettra de combler ce manque en y ajoutant des fonctionnalités que nous jugeons importantes.

*GazeTracker*TM ne se contente pas d'être un outil de création d'expérimentations, il est surtout utile par le fait qu'il permet d'extraire les informations récoltées lors de l'expérimentation dans des fichiers. Pour chaque sujet ayant passé l'expérience, un fichier contenant toutes ces informations est créé. Plus précisément, ce fichier contient les données oculaires du sujet, c'est-à-dire que pour chaque fixation/saccade une ligne est créée. Ces différentes lignes contiennent le nom de l'image, une durée en secondes (avec une précision à la milliseconde), le code indiquant s'il s'agit d'une saccade ou d'une fixation, deux coordonnées x et y, le diamètre de la pupille en largeur et en hauteur ainsi qu'un numéro d'identification.

Un tel oculomètre détecte le regard exclusivement via son système infrarouge. Cette contrainte oblige donc à placer d'une manière judicieuse les caméras vis-à-vis des yeux, plus particulièrement vis-à-vis des yeux *cachés* derrière des lunettes. Suite à un problème de reconnaissance des yeux lors de nos tests causé par la présence d'un reflet lumineux, nous nous sommes dirigés, à cause d'une mauvaise connaissance du matériel, vers une solution complètement erronée. En effet, nous avons essayé de jouer avec la luminosité de la pièce afin de faire disparaître ce reflet. Or, ce reflet provenait de la source infrarouge et servait donc de repère pour les caméras. Une fois le concept compris, nous avons pu placer les caméras et la source infrarouge sous un certain angle afin d'optimiser l'efficacité de l'oculomètre.

7.2 TAUPE

Pour travailler avec l'eye-tracker, nous avons hérité du logiciel TAUPE (*Thoroughly Analysing the Understanding of Programs through Eyesight*). Ce logiciel a été entièrement conçu par le laboratoire Ptidej. À noter que le logiciel TAUPE est sous la licence GNU GPL v3, ce qui signifie qu'il est entièrement libre.

TAUPE est un logiciel qui n'a cessé d'évoluer. Afin de donner une idée de cette évolution, nous présentons rapidement différentes métriques qui ont été calculées à l'aide du *plug-in* Eclipse nommé Metrics² en tableau 7.1 ainsi qu'un bref historique de ces différentes versions :

2. <http://metrics.sourceforge.net/>



FIGURE 7.1 – L’oculomètre FaceLAB de l’École Polytechnique de Montréal

	v0	v1	v2.0	v2.1
Nombre total de lignes de code	812	4912	12238	16423
Nombre de Packages	3	11	25	28
Nombre de Classes	10	71	145	180
Nombre d’Interfaces	0	10	7	7
Nombre de Méthodes	40	490	800	998
Nombre d’Attributs	30	248	333	453
Manque de cohésion des méthodes	0.329	0.37	0.225	0.213

Tableau 7.1 – Évolution des métriques de TAUPE en fonction des versions

- La **version initiale** permettait simplement d’importer les données d’un seul système d’oculométrie (Eye-link II) et proposait un système simple de visualisation des fixations et des saccades. Il a été écrit par Yann-Gaël Guéhéneuc.
- **TAUPE 1.0** a été écrit et maintenu par plusieurs étudiants en stage (principalement par Bertrand Van Den Plas) au *Ptidej Lab*. La version 1.0 permettait à l’utilisateur d’appliquer plusieurs algorithmes sur les données, notamment un algorithme de correction d’*offset* statique.
- **TAUPE 2.0** a été réalisée en 2010-2011. Durant cette mise à jour, la totalité du logiciel a été ré-écrite par Lorent Lempereur et Benoit De Smet. Cette nouvelle version est capable de gérer plusieurs types de fichiers générés par plusieurs systèmes d’oculométrie différents et contient de nouvelles fonctionnalités.
- **TAUPE 2.1** est la dernière version du logiciel (2011-2012). Comme détaillé plus bas, notre contribution sur TAUPE se concentre sur un ajout de fonctionnalités ainsi qu’une amélioration de celles existantes.

Afin de ne pas alourdir inutilement ce chapitre, nous détaillerons uniquement les versions avec lesquelles nous avons été directement liées. Ainsi, la **version 2.0** représente pour nous le point de départ de ce logiciel et la **version 2.1** montre les nouveautés que nous avons apportées.

7.2.1 TAUPE v2.0

Initialement, TAUPE avait été écrit pour comparer les différentes façons d’observer et d’analyser des diagrammes UML. En effet, le but premier de TAUPE était de pouvoir analyser les mouvements

des yeux. Mouvements qui sont enregistrés par un oculomètre durant un questionnaire portant sur la compréhension de diagrammes. Dans sa version 2.0, TAUPE a dépassé cet objectif premier en proposant d'autres fonctionnalités.

L'architecture conceptuelle

L'architecture de TAUPE est représentée par la figure 7.2. L'architecture proposée dans la version 2.0 reprend l'ensemble des éléments en noir³. Le lecteur remarquera sans mal que l'architecture de la version 2.0 reste dominante dans la version actuelle. C'est pour cette raison que nous détaillons l'architecture dans cette section.

Comme décrit dans la figure 7.2, les données d'entrées principales de TAUPE sont des *fichiers sujets*. Plus concrètement, chaque sujet qui passe l'expérience aura deux fichiers qui lui sont propres. Un fichier reprend ses données oculométriques et l'autre s'intéresse à des sujets plus généraux (sexe, connaissance d'un domaine particulier, niveau d'étude, etc).

Comme nous l'avons vu dans la section 7.1, l'École Polytechnique de Montréal dispose de deux eye-tracker. Cependant ceux-ci n'enregistrent pas les données de sorties dans des fichiers de même type. En effet, EyeLinkII enregistre ses résultats dans un format « edf »⁴ alors que FaceLAB/GazeTracker propose un fichier en format « out ». Dès lors, TAUPE a été muni de deux *parsers*. Le rôle de ces *parsers* est capital dans le sens où ils permettent d'avoir les *saccades* et les *fixations* du sujet.

Enfin, les derniers fichiers pris en considération sont les différentes questions, les *fichiers questions*, qui ont été posées au sujet. Ces questions sont représentées par un ensemble d'images et leurs zones d'intérêt correspondantes.

Une fois tous ces fichiers chargés, TAUPE propose toutes sortes de *commandes*. La commande *AOIMaker* concerne les fichiers questions, et plus particulièrement la création des zones d'intérêt, une commande existe afin de les créer via le logiciel. Cependant, la commande la plus importante de TAUPE est la possibilité d'exécuter des *algorithmes* sur les données et de produire des résultats. Ces résultats peuvent être écrits dans des fichiers en utilisant des *printers* choisis par l'utilisateur. Et en dernier lieu, la commande *CorrectionOffset* permet de corriger les offsets directement sur le fichier sujet.

Les fonctionnalités

Comme dit précédemment, TAUPE permet bien plus que d'analyser les données récoltées avec un oculomètre. En effet, le logiciel possède également d'autres fonctionnalités :

- **Un outil de visualisation des données oculométriques.** La particularité de cet outil est qu'il propose une multitude d'options (fixations, saccades, zones d'intérêt/non-intérêt,...) pour filtrer le contenu qui apparaîtra à l'écran. Via ce système, l'utilisateur peut facilement afficher les informations qui lui semblent les plus pertinentes. Cet outil comprend aussi un système de pourcentage de fixations. C'est à dire que l'utilisateur peut choisir le pourcentage de fixations à regrouper pour l'affichage. Le regroupement de fixations permet de regrouper deux fixations consécutives qui se trouvent dans une même zone. La durée de la fixation regroupée est la

3. Les éléments en rouge sont des ajouts de la version 2.1.

4. D'ailleurs, dans un soucis technique, ses données doivent être converties en XML par l'outil `edf2xml`.

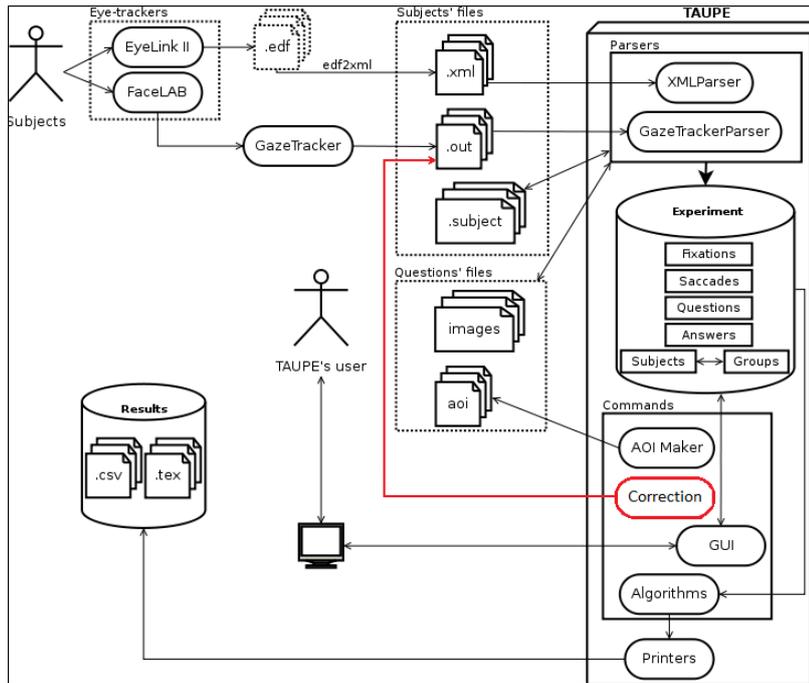


FIGURE 7.2 – Architecture conceptuelle de TAUPE

somme des durées des fixations qui la composent.

Dans cet outil, chaque élément qui peut être affiché possède une couleur attribuée. Cette couleur est explicitement choisie par l'utilisateur, ce qui lui permet de personnaliser son diagramme. De plus, le diagramme affiché dépend, bien évidemment, du sujet qui a passé l'expérience ainsi que la question du sujet. Ces deux éléments peuvent être sélectionnés via les listes déroulantes situées en haut à droite de la figure.

- Un outil de création des zones d'intérêt et/non-intérêt sur une image.** Cette fonctionnalité, appelée *AOIMaker*, permet de sélectionner une image (principalement des graphiques dans notre cas) et d'y créer des zones d'intérêt qui seront exportées dans un fichier de zone d'intérêt. La création de ces zones se fait exclusivement par des clics de souris successifs. Une vérification de cette zone est ensuite réalisée pour être sûr que la zone créée est bel et bien un *polygone de Jordan*. Pour chaque zone, l'utilisateur donne un nom, un identifiant et détermine « son intérêt ». Une zone peut être qualifiée suivant trois « intérêt ». Ceux-ci sont « *intérêt* », « *non-intérêt* » ou « *null* ». Les deux premiers sont assez explicites, le dernier considère la zone comme ne faisant pas partie de l'image. Dès lors, toutes les fixations s'y trouvant sont ignorées pour les différents calculs et/ou algorithmes. Ces fichiers de zone d'intérêts peuvent également être écrits manuellement mais doivent respecter la grammaire suivante (décrite en *Extended Backus-Naur Form*) :

```
[<id> <type> <name> <coordonate> <coordonate> <coordonate>+ <EOL>]*
<id> ::= integer
<type> ::= NULL | AOI | AORI
<coordonate> ::= "(" integer "," integer ")"
```

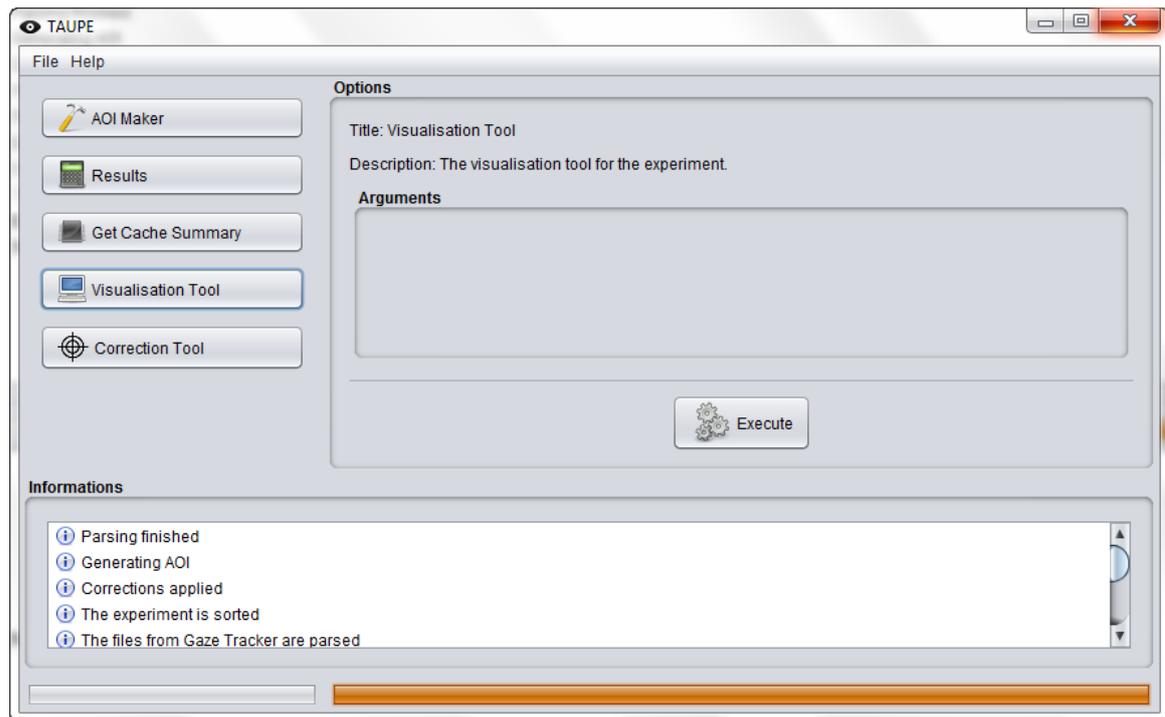


FIGURE 7.3 – Fenêtre principale du logiciel TAUPE

<EOL> ::= EndOfLine

- **Un outil d'exportation des données.** Les formats d'exportation disponibles sont les formats "LATEX" ou "csv". Le choix de l'exportation est laissé à l'utilisateur. Ces données peuvent, par la suite, être directement exploitées par un logiciel de *datamining* comme le logiciel "R".
- TAUPE doit implémenter **des analyseurs syntaxiques** afin de récupérer les données des fichiers générés par les oculomètres. Étant donné qu'il existe de nombreux systèmes d'oculométrie différents, il doit être facile d'ajouter de nouveaux analyseurs syntaxiques au programme. Cette fonctionnalité permet à TAUPE de ne pas dépendre du matériel utilisé.

7.2.2 TAUPE v2.1

Dans sa version 2.1, l'architecture et le fonctionnement de TAUPE n'ont pas été beaucoup modifiés. En effet, seuls les éléments en rouge sur la figure 7.2 représentent les éléments ajoutés à cette architecture. On peut donc en déduire que la version actuelle se base presque entièrement sur la version 2.0 (décrit en section 7.2.1). Lors de cette mise à jour, l'accent a été mis sur une amélioration générale de TAUPE. Dès lors, le travail réalisé se porte plus au niveau des fonctionnalités manquantes ou de manque d'utilisabilité de certaines fonctionnalités, que sur une refonte du logiciel⁵.

Dans cette dernière version, TAUPE propose cinq fonctionnalités principales comme illustré dans la figure 7.3 : « *OIAMaker* », « *Result* », « *GetCacheSummary* », « *Visualisation Tool* » et « *Correction Tool* »

Nouvelles métriques

Certaines métriques citées par des travaux antérieurs aux été ajoutées par nos soins. La liste exhaustive de ces métriques est :

- Pourcentage de transition AORI vers AOII

5. Raison pour laquelle, nous parlons ici d'une version 2.1 et non pas d'une version 3.0.

- Ratio de la durée des fixations et des saccades
- Nombre moyen de fixations Post-AORI
- Dynamic Time Warping

L'intérêt d'avoir ajouté ces métriques est qu'elles permettent d'apporter de nouvelles réponses quant aux comportements visuels des sujets. Les détails de ces différentes métriques peuvent être trouvés en section 4.3.3.

AOIMaker : amélioration de l'utilisabilité

L'outil de création des zones d'intérêt et/non-intérêt sur une image a été repensé. La création d'une zone d'intérêt reste la même mais quelques nouveautés ont été implémentées.

Premièrement, dans la version 2.0 de TAUPE, l'utilisateur n'avait pas droit à l'erreur dans la création de sa zone d'intérêt. En effet, il n'était pas possible de retirer un point de son polygone (qui définit la zone d'intérêt). Dès lors, une nouvelle méthode a été implémentée afin que l'utilisateur puisse faire « marche arrière » lors de la mise en place de cette zone. Pratiquement, nous avons opté de garder le *clic gauche* pour ajouter un point du polygone tandis que le *clic droit* permet de lui retirer un point.

Deuxièmement, nous avons voulu optimiser la création des fichiers « .aoi ». Les expériences d'eye-tracking utilisant comme support des diagrammes de classes possèdent un avantage qui n'était pas encore utilisé. En effet, dans de telles expériences, chaque représentation se retrouve plusieurs fois dans l'expérience. Concrètement, une représentation d'un framework se retrouvera « x » fois dans l'expérience suivant les « x » questions posées sur ce framework. Bien sûr, les zones d'intérêt pour un diagramme de classes sont souvent les classes elles-mêmes. Ainsi, nous avons décidé d'autoriser l'utilisateur à éditer les zones d'intérêt au niveau de leur nom et de leur « intérêt ». L'avantage réside dans le fait que si l'utilisateur charge un fichier de zone d'intérêt (*Q1.aoi*), il peut créer facilement un autre fichier de zone d'intérêt (*Q2.aoi*) qui reprend une grande partie des zones de *Q1.aoi* mais qui ne donne pas les mêmes niveaux d'intérêts. Cette similitude des zones est illustrée par la figure 7.4. Par cette fonctionnalité, on évite à l'utilisateur de créer plusieurs fois des zones identiques.

Outil : correction des offsets

La plus grande lacune du logiciel TAUPE était, sans aucun doute, l'absence d'outil permettant de corriger les offsets dûs aux erreurs d'enregistrement des eye-trackeurs.

L'outil de correction des offsets étant une nouvelle fonctionnalité à part entière, nous l'avons intégrée à une nouvelle *commande* dans l'architecture du logiciel. Il existe deux sortes d'offsets, les statiques et les non statiques. Afin de répondre au mieux à ces différents offsets, l'outil intègre une correction d'offsets statiques ainsi qu'une correction d'offsets non statiques. Le principe de cet outil est de « déplacer » les fixations afin de les positionner à l'endroit où elles auraient dû être.

Pour les offsets statiques, l'outil demande une correction globale de l'image. L'utilisateur rentre deux nombres. Ces deux valeurs sont, respectivement, les modifications apportées à l'axe x et à l'axe y .

Pour la correction des offsets non statiques, l'outil permet à l'utilisateur de dessiner un polygone, représentant la zone des fixations à corriger⁶. Une fois la zone créée et verrouillée par l'utilisateur, le programme retient les fixations comprises dans cette zone. Ensuite, la procédure renvoie à la correction des offsets statiques mais uniquement pour les fixations sélectionnées.

Dans tous les cas, une vérification est effectuée avant tout changement pour s'assurer qu'aucune

6. Cette méthode permet également d'éditer la figure, c'est-à-dire de supprimer un point dans le polygone.

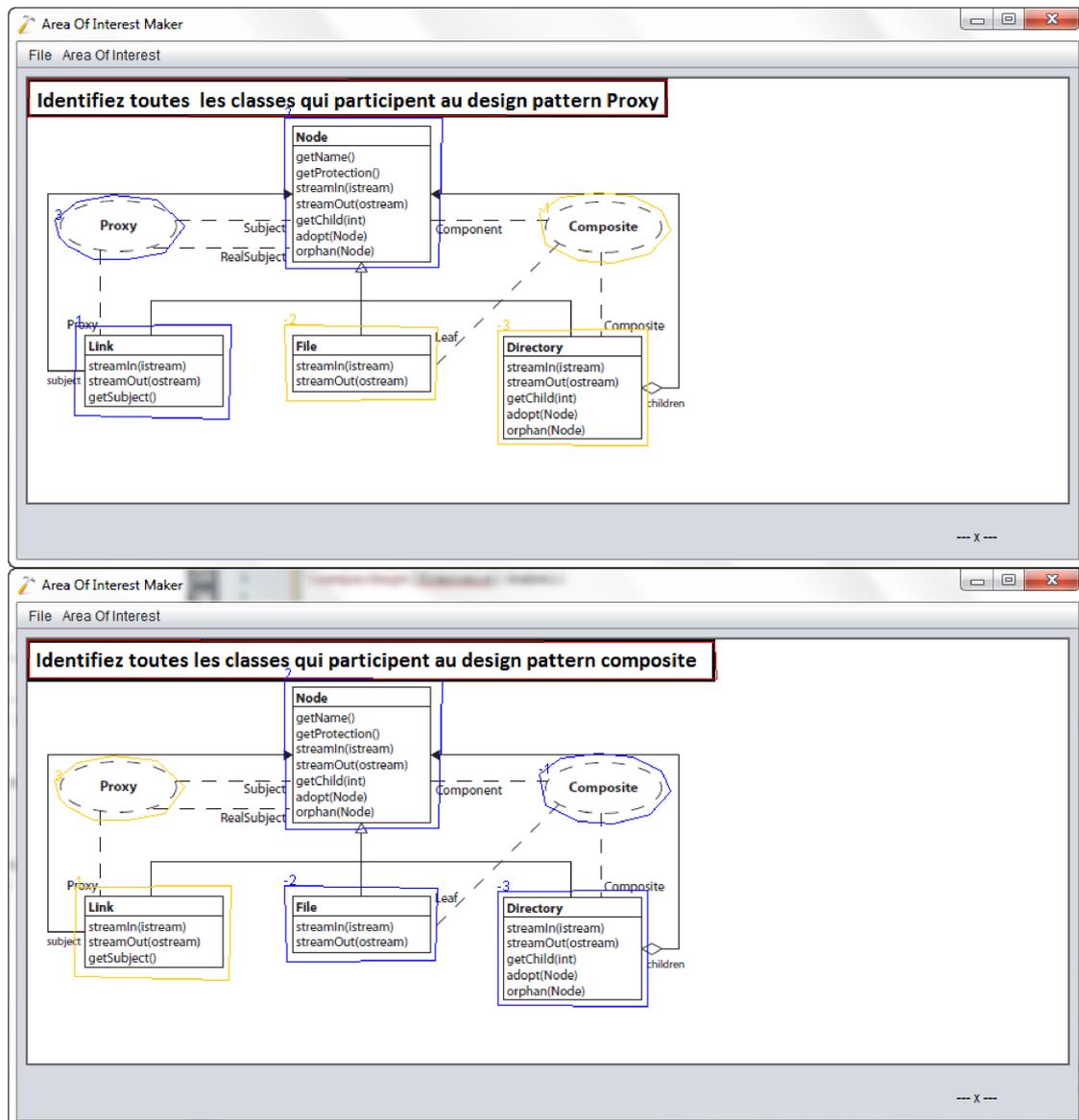


FIGURE 7.4 – Zones d'intérêt et de non intérêt de la question « Q1 » et de la question « Q2 »

fixation ne sorte de l'image. Dès que les corrections sont effectuées, l'utilisateur peut choisir entre enregistrer les nouvelles fixations sur le fichier déjà existant ou les enregistrer dans un nouveau fichier.

Heatmap

Les **heatmap** représentent une indication intéressante dans les expérimentations, et plus particulièrement dans les expériences d'eye-tracking. En effet, par son fonctionnement, l'heatmap permet de donner une vision globale mais efficace de l'intérêt qu'a porté une personne sur un diagramme. Cette nouvelle fonctionnalité utilise les fixations présentes sur l'image afin d'en dessiner un spectre transparent qui, suivant sa couleur, signifie si la zone a été souvent regardée ou non. Plus concrètement, le fonctionnement principal des heatmap repose sur le fait que chaque pixel de l'écran reçoit une valeur attribuée par le comportement de l'utilisateur (dans notre cas, ce comportement est représenté par les différents points, nommés fixations, que l'utilisateur a regardés). Ensuite, chaque valeur est associée à une couleur suivant une échelle pré-établie, ce qui donne le spectre de couleur [WF08].

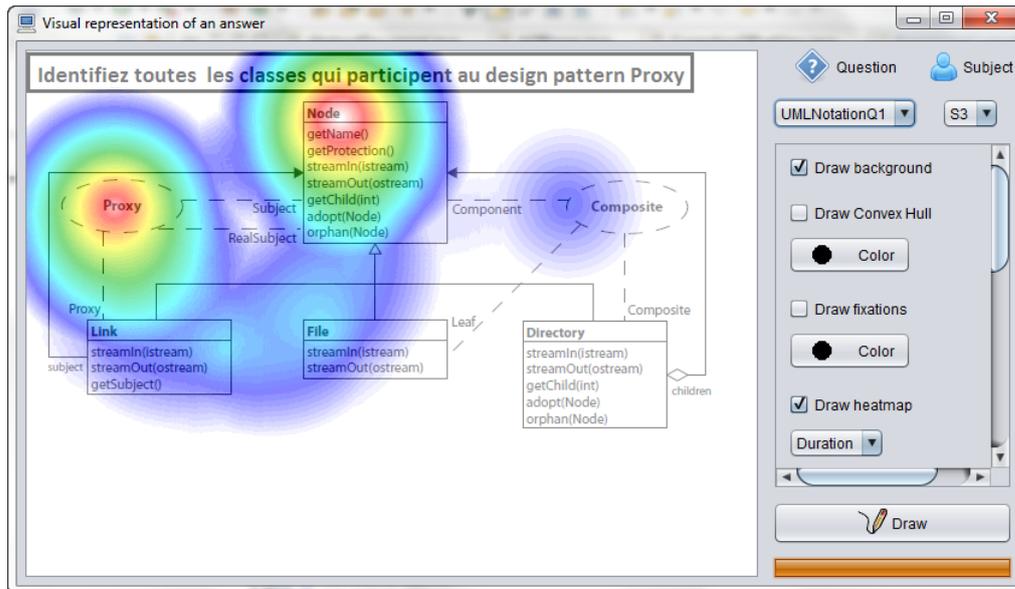


FIGURE 7.5 – Heatmap basée sur la durée des fixations

Deux versions ont été implémentées dans TAUPE. La première version, nommée **density**, donne le même ratio à toutes les fixations. Ce qui signifie qu'un pixel ne reçoit une valeur élevée qu'au cas où il y aurait plusieurs fixations à sa proximité.

Tandis que la deuxième version, nommée **duration**, fonctionne comme son nom l'indique, sur les durées des fixations. Concrètement, elle donne une valeur « *d'influence* » plus importante aux fixations plus longues et une valeur plus faible aux fixations plus courtes.

La figure 7.5 montre une heatmap basée sur la durée des fixations. La couleur verte indique que la zone a été peu sollicitée tandis que le rouge montre les zones que le sujet a fixées le plus longtemps.

L'intégration des heatmap a été effectuée au niveau de « l'outil de visualisation des données oculométriques ». L'ajout d'un nouvel outil adressé uniquement aux heatmap n'aurait pas été optimal. De plus, l'outil de visualisation, offrant une multitude d'options d'affichage, permet d'optimiser l'utilisation de ces heatmap.

Distance relative entre chemins visuels

Ce chapitre présente une métrique qui s'intéresse à la dynamique des parcours visuels. L'intérêt de cette métrique est qu'elle se différencie des autres. En effet, bien que toutes les métriques présentées dans la section 4.3.3 soient intéressantes, aucune d'entre elles ne compare réellement les parcours visuels des sujets. Le but de cette métrique, nommée distance relative entre chemins visuels, est de comparer des parcours visuels deux-à-deux et de déduire par un algorithme une valeur qui quantifie la différence entre ces deux parcours. Plus la valeur de cette métrique sera élevée, plus la différence entre deux parcours visuels sera grande.

La section 8.1 explique brièvement l'intérêt de cette métrique ainsi que les méthodes utilisées pour la calculer. La section 8.2 s'intéresse plus particulièrement à la méthode basée sur la distance de Levenshtein tandis que la section 8.3 présente l'approche qui s'inspire sur l'algorithme que l'on nomme Dynamic Time Warping. Enfin, ce chapitre se clôture par une étude de cas mettant en confrontations ces deux approches en section 8.4

Sommaire

8.1	Principe	72
8.2	Distance de Levenshtein	72
	8.2.1 Définition	72
	8.2.2 Limites	73
	8.2.3 Algorithme	73
8.3	Dynamic Time Warping	73
	8.3.1 Définition	73
	8.3.2 Technique	75
	8.3.3 Algorithme	77
	8.3.4 Avantages	77
8.4	Étude de cas	77

8.1 Principe

La métrique **distance relative entre chemins visuels** peut être utilisée afin d'estimer et de quantifier la différence entre deux parcours visuels et peut ouvrir la voie à de nouvelles métriques se concentrant sur la dynamique du parcours et non plus seulement sur son aspect statique.

Jusqu'à peu, il n'existait pas réellement de telle métrique dans la littérature, et ce, même dans le domaine de l'oculométrie. La difficulté d'établir cette métrique réside dans le fait de la rendre viable et utile. Cependant, les travaux de De Smet et de Lempereur [DL11] ont amené la première métrique de distance relative entre chemins visuels. Celle-ci se base sur la distance de Levenshtein.

Les auteurs de cette méthode, bien que satisfaits de leur travail, ont énoncé leurs craintes vis-à-vis de celle-ci dans certaines situations. C'est pour cette raison que nous avons créé une autre méthode pour calculer cette métrique. Celle-ci se base sur l'algorithme *Dynamic Time Warping*.

Ces approches sont présentées dans la suite de ce chapitre avec leurs forces et leurs faiblesses. Pour les deux méthodes présentées, nous avons considéré les fixations comme étant la base des chemins visuels. En effet, les fixations représentent les données les plus fiables et permettent facilement de construire ce chemin visuel. De plus, le fait qu'elles soient enregistrées chronologiquement représente un avantage non négligeable.

8.2 Distance de Levenshtein

8.2.1 Définition

La première solution adoptée dans TAUPE provient de De Smet et Lempereur [DL11, p. 102]. Leur travail se base essentiellement sur des algorithmes qui permettent de calculer la distance entre deux chaînes de caractères (*string*), nommée *edit distance* ou *distance de Levenshtein*.

L'*edit distance* entre deux chaînes de caractères se concentre sur la transformation d'un *string* en un autre par une série d'opérations d'édition sur des caractères individuels. Les opérations permises sont [Gus97, p. 215] :

- L'**insertion** I d'un caractère dans le premier *string*.
- La **suppression** S d'un caractère dans le premier *string*.
- La **substitution** (ou remplacement) R d'un caractère dans le premier *string* par un caractère du second.

Gusfield définit :

“L'*edit distance* entre deux chaînes de caractères est définie telle que le minimum d'opérations d'édition – insertion, suppression ou substitution – nécessaires pour transformer la première chaîne de caractères en la seconde.” [Gus97]

L'*edit distance* est parfois référée comme distance de Levenshtein. Sa définition implique que toutes les opérations sont faites sur une seule chaîne de caractères. Néanmoins, l'*edit distance* est parfois considérée comme le nombre minimum d'opérations faites sur les deux chaînes pour transformer les deux en une troisième chaîne commune. Cette vision est équivalente à celle définie ci-dessus, car une insertion dans un *string* peut être vue comme la suppression dans l'autre et vice-versa [Gus97, p. 216].

8.2.2 Limites

Le fait de ne comparer que des string réduit l'intérêt de cette métrique. Comme énoncé dans leur travail, les auteurs se sont rendus compte de plusieurs problèmes inhérents au problème de réduction.

La première difficulté qui se pose est de déterminer comment sélectionner les éléments qui serviront à comparer les deux parcours visuels? Si il est clair que l'algorithme compare deux chaînes de caractères, l'établissement de ces chaînes pose encore problème. Il est vital qu'un certain niveau de granularité soit adopté par la métrique. Les auteurs ont privilégié la piste des zones d'intérêt afin de déterminer les différents éléments qui serviront de base pour ces chaînes. Ce choix est motivé par l'importance que donne la théorie de la vision à ces zones d'intérêt. Ils définissent donc un parcours visuel comme une suite d'éléments, sous entendu de zones d'intérêt ou de non intérêt, parcourue chronologiquement par le regard d'un sujet.

Ce choix de ne prendre que les zones d'intérêt entraîne directement une nouvelle difficulté. Que faire si, dans certains cas, le sujet est amené à visualiser plusieurs fois une même zone d'intérêt? La solution adoptée par les auteurs s'adapte suivant les cas. Ainsi, si une zone d'intérêt est visitée deux fois mais que le parcours visuel est passé par une autre zone d'intérêt entre temps, alors les deux passages dans la même zone sont considérés comme différents. Si ces deux passages dans une même zone sont successifs, alors ils sont considérés comme identiques. Ce choix se justifie de la façon suivante : *un sujet qui maintient son regard dans une zone sans la quitter s'y intéresse une fois, peu importe le nombre de fixations qui correspondent à cet intérêt*. Ce choix implique tout de même une perte de précision. Dans cette métrique, deux sujets qui passent par les mêmes zones d'intérêt auront des chemins visuels semblables alors que la réalité peut être différente. Ce danger est représenté dans la figure 8.1 où le premier sujet, dont la couleur est rouge, s'intéresse plus à la classe « Node » tandis que le deuxième sujet, représenté en bleu, parcourt d'avantage la classe « File ». Or le sujet en rouge est plus efficace, car ses fixations consécutives sont dans une zone d'intérêt alors que le sujet bleu perd plus de temps sur une zone de non intérêt. Malheureusement, avec la méthode mise en place par De Smet et Lempereur, la différence d'efficacité présente entre ces deux parcours visuels ne sera pas relevée. La seconde méthode présentée, dans la section suivante, tient compte de ce genre de situation.

8.2.3 Algorithme

L'algorithme permettant de calculer cette distance est détaillé en pseudo-code dans l'Algorithme 1 tel que présenté par Wagner et Fischer [WF74]. C'est le résultat de cet algorithme appliqué à notre représentation du parcours visuel que nous avons nommé **distance relative entre chemins visuels**. Pour des raisons pratiques, cet algorithme a été modifié par De Smet et Lempereur afin de pouvoir être utilisé par le logiciel TAUPE. Pour se faire, ils ont remplacé les deux chaînes de caractères par deux listes d'objets. Ensuite, ils ont établi une relation d'équivalence entre ces objets.

8.3 Dynamic Time Warping

8.3.1 Définition

Dynamic Time Warping, que l'on réduit par son acronyme « DTW », est un système de mesure de distance entre deux courbes. DTW se base sur les différents points qui composent les courbes afin de les comparer. Ce paragraphe aborde plusieurs notions afin de comprendre comment fonctionne

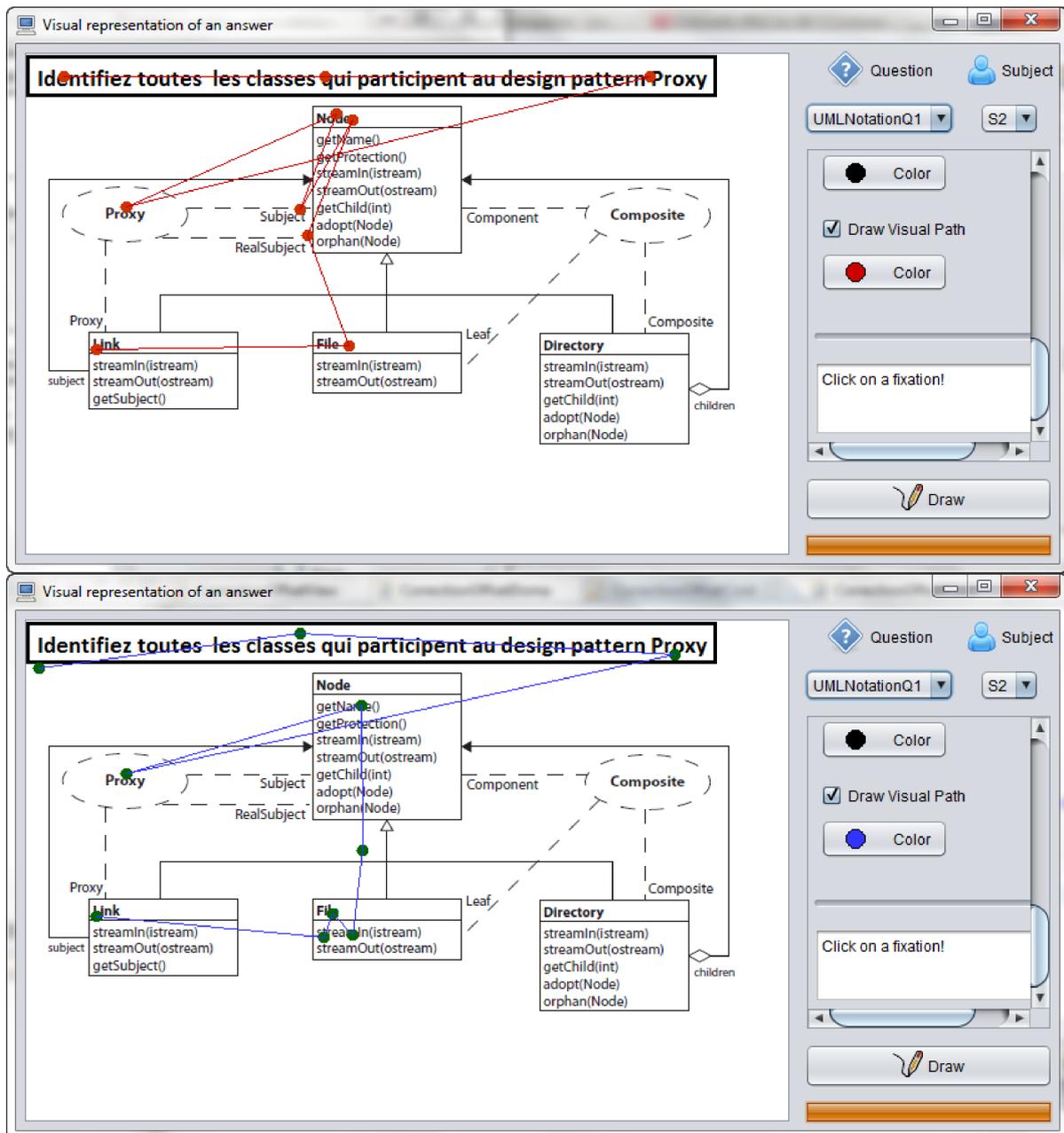


FIGURE 8.1 – Comparaison entre deux chemins visuels

réellement ce système. Ainsi, une brève introduction à l'élaboration de DTW et les conditions que demande ce système seront abordées. Un aperçu de l'algorithme est également présenté au lecteur, sous forme de pseudo-code.

Pour pouvoir comparer deux courbes, DTW a besoin que ces courbes soient « discrètes »¹ et « online »²[Nie04]. Ces deux conditions sont respectées par le choix de prendre les fixations comme base afin d'établir les courbes. En effet, par leur nature, les fixations assurent la condition *discrète*. De

1. *Discrète* signifie que la courbe est composée d'un nombre limité de points, comme elle pourrait être représentée dans un ordinateur. A l'opposé de *continue* qui signifie que la courbe est constituée d'un trait continu, comme elle pourrait être représentée dans le monde réel. Notez que la représentation discrète détient moins d'information que la représentation continue.

2. *Online* signifie les informations contiennent des données sur le temps. A l'opposé d' *offline* qui signifie qu'aucune information sur le temps n'est disponible.

Algorithme 1: Algorithme de calcul de l'*edit distance* entre deux chaînes de caractères

```

Entrées :
s: char[1..m] la première chaîne de caractères où  $m \geq 0$ 
t: char[1..n] la première chaîne de caractères où  $n \geq 0$ 
Résultat : La valeur ( $\in \mathbb{N}^+$ ) de l'edit distance de s et t

d  $\leftarrow$  new integer[0..m,0..n]
for i  $\leftarrow$  0 to m do
  | d[i,0]  $\leftarrow$  i;
end
for j  $\leftarrow$  0 to n do
  | d[0,j]  $\leftarrow$  j;
end
for j  $\leftarrow$  1 to n do
  | for i  $\leftarrow$  1 to m do
    | if s[i] = t[j] then
      | d[i,j]  $\leftarrow$  d[i-1,j-1];
    | else
      | d[i,j]  $\leftarrow$  minimum(d[i-1,j]+1,d[i,j-1]+1,d[i-1,j-1]+1);
    | end
  | end
end
return d[m,n];

```

plus, chaque fixation possède une valeur qui représente à quel moment elle a été enregistrée ce qui donne à la courbe la deuxième condition.

8.3.2 Technique

Une technique pour faire correspondre les chemins suit essentiellement les étapes suivantes afin de calculer la distance entre ces deux courbes :

1. Un chemin de correspondances doit être créé. Un chemin de correspondances est une liste de combinaisons des points de la première courbe avec les points de la seconde courbe. La technique utilisée pour la création de cette liste est ce qui distingue les différentes méthodes, dont Dynamic Time Warping fait partie.
2. Pour chaque combinaison des fixations venant de la première courbe (fixation i) et de la deuxième courbe (fixation j) dans le chemin de correspondances, la distance, nommée $D(i,j)$, entre elles est calculée. Plusieurs méthodes pour calculer la distance entre deux points existent. Cette implémentation de DTW utilise la distance euclidienne. La distance euclidienne entre deux points $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ est calculé par $(|PQ| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2})$
3. Les distances calculées dans les étapes précédentes sont additionnées et cette distance totale est normalisée en la divisant par le nombre de combinaisons présentes dans le chemin de correspondance. Le résultat de cette division correspond à la distance DTW des deux courbes.

La figure 8.2 montre une matrice qui représente un chemin de correspondance entre deux courbes de longueurs différentes. Par exemple, on peut voir que le point 3 de la première courbe est associé au point 2 de la deuxième courbe.

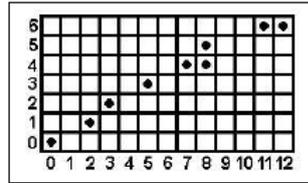


FIGURE 8.2 – Chemin de correspondances entre deux courbes dont les longueurs sont différentes

DTW construit un chemin de correspondances avec une manière assez complexe. Il est basé sur *une correspondance linéaire*³ mais d'autres conditions ou contraintes sont ajoutées. Ces conditions sont appelées *la condition de continuité*, *la condition des frontières* et *la condition de monotonie*. Deux de ces conditions sont facultatives et elles ont toutes un certain nombre d'options qui peuvent être changées. [Nie04]

Condition de continuité :

La condition de continuité décide à quel point le chemin de correspondances est autorisé à différer de la correspondance linéaire. Cette condition est au cœur de DTW et n'est donc pas facultative. Cette condition est formulée par Vuori *et al.* [VAL+00] telle que si N_1 et N_2 sont le nombre des points de la première courbe et de la deuxième courbe, respectivement, le $i^{\text{ème}}$ points de la première courbe et le $j^{\text{ème}}$ points de la deuxième courbe peuvent correspondre si $N_2/N_1 * i - c * N_2 \leq j \leq N_2/N_1 * i + c * N_2$. Le paramètre c détermine si ce chemin de correspondances va différer de la correspondance linéaire ou pas. Si $c = 1$, la condition de continuité n'a aucun effet (c'est-à-dire que tous les points de la première courbe peuvent être associés avec tous les points de la seconde courbe). Dans ce cas, le chemin de correspondances peut différencier totalement de la correspondance linéaire. Au contraire, si $c = 0$, la condition est tellement forte que le chemin de correspondances qui en résulte est exactement le même que la correspondance linéaire.

Condition des frontières :

La condition des frontières force la correspondance entre le premier point de la première courbe avec le premier point de la deuxième courbe. Il en va de même pour le dernier point. A noter que ces points seront bien associés même si la condition de continuité n'est pas d'accord avec ces correspondances.

Condition de monotonie :

La condition de monotonie est une condition qui interdit que des correspondances se fassent en « revenant en arrière ». Si à un moment donné dans le processus de correspondances, il est décidé que le point i de la première courbe est associé avec le point j de la seconde courbe, il n'est plus possible pour n'importe quel point de la première courbe postérieur à i d'être associé à un point de la seconde courbe antérieur à j . De même, il est également interdit d'avoir une correspondance entre un point de la première courbe antérieur à i et un point de la seconde courbe postérieur à j . La figure 8.3 illustre ces propriétés mathématiques.

3. Une façon de créer un chemin de correspondances est appelée correspondance linéaire. Cette correspondance ne peut être utilisée que si les deux courbes sont de longueur égale (le même nombre de points) : chaque point i de la première courbe est associé avec le point j de la deuxième courbe. Une version un peu plus sophistiquée de cette technique rend possible la correspondance de deux courbes possédant des longueurs différentes. Si N_1 et N_2 sont le nombre de points dans les deux courbes, le point i de la première courbe et le point j de la deuxième courbe 2 correspondent si $i - 1/N_1 * N_2 \leq j \leq i/N_1 * N_2$.

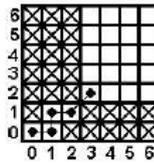


FIGURE 8.3 – Chemin de correspondances où est appliquée la condition de monotonie. Comme on le voit, aucun retour en arrière n’est permis

8.3.3 Algorithme

L’algorithme permettant de calculer cette distance est détaillé en pseudo-code dans l’Algorithme 2 tel que présenté par Ralph Niels [Nie04]. Son Algorithme s’appliquait initialement à des courbes représentant des lettres. Une légère adaptation des données d’entrées a été nécessaire, en remplaçant les points qui forment la structure des lettres par les fixations des chemins visuels, afin que cet Algorithme calcule bien la distance entre deux chemins visuels.

8.3.4 Avantages

Les principaux avantages de l’utilisation de Dynamic Time Warping par rapport à la distance de Levenshtein sont :

- Avec la distance de Levenshtein, on comparait les chemins par rapport aux zones d’intérêt... Cette imprécision pouvait donner des erreurs. Avec l’algorithme de DTW, la comparaison des différents chemins se fait en prenant les coordonnées des différents points, c’est-à-dire les fixations.
- Grâce au fait que DTW ne prend plus en compte les zones d’intérêt, ce système permet d’enlever une contrainte. En effet, la limitation qui imposait de ne pas prendre les fixations successives d’une zone d’intérêt n’existe plus.
- Un problème lié à la distance de Levenshtein est qu’il se résume à différencier les chemins suivant leurs chaînes de caractères qui les représentent. Or, des courbes n’ayant pas la même taille donnaient une distance assez grande due à ce calcul. Avec DTW, deux courbes qui se ressemblent mais qui ont des tailles différentes auront une distance plus petite.
- L’algorithme DTW s’applique facilement aux chemins visuels, et est donc optimisé pour une telle recherche.

Cependant le système DTW ne permet pas, comme pour la distance de Levenshtein, de calculer la distance absolue des chemins visuels. En effet, il ne s’agit que d’une distance relative.

8.4 Étude de cas

Comme nous venons de le voir au cours des deux sections précédentes, l’approche de la distance de Levenshtein et l’approche de DTW peuvent être utilisées pour calculer une distance relative entre deux chemins visuels. Le principe des deux méthodes nous permet de comparer le résultat de celles-ci en considérant un ensemble de parcours visuels de base. Ainsi, cette comparaison a pour but de vérifier l’équivalence des deux méthodes.

Une étude de cas est présentée ci-dessous afin de vérifier les différences réelles entre ces deux techniques. Pour réaliser ce travail, nous avons repris les données de l’expérience menée par De Smet

Algorithme 2: Algorithme de calcul de *Dynamic Time Warping* entre deux liste de points

```

Entrées :
list1 : Liste des fixations de la la première courbes où list1 != null
list2 : Liste des fixations de la la deuxième courbes où list2 != null
Résultat : La valeur ( $\in \mathbb{N}^+$ ) de la distance DTW de list1 et list2

foreach  $i \in list1$  do
   $smallestDistance \leftarrow +\infty$ ;
  foreach  $j \in list2$  do
    if ((boundary condition = on) AND (( $i=0$  AND  $j=0$ ) OR ( $i=list1$  AND
       $j=list2$ ))) then
      add  $i$ - $j$  combination to matching path;
      add distance between  $i$  and  $j$  to total distance;
    else
      if ( $i$  matches  $j$  then
        if ( $distance\ i-j < smallest\ distance$ ) then
           $smallestDistance = distance$  between  $i$  and  $j$ ;
        end
      end
    end
  end
  if ( $i$ - $j$  combination with smallest distance is not in path yet) then
    add  $i$ - $j$  combination with smallest distance to matching path;
    add smallest distance to total distance;
  end
end

/*The next part of the heuristic guarantees that the DTW-distance is symmetrical*/

foreach  $j \in list2$  do
   $smallestDistance \leftarrow +\infty$ ;
  foreach  $i \in list1$  do
    /*if the boundary condition is on, it already is satisfied by the first part of the
    heuristic*/ if ( $j$  matches  $i$  then
      if ( $distance\ j-i < smallest\ distance$ ) then
         $smallestDistance = distance$  between  $j$  and  $i$ ;
      end
    end
  end
  if ( $j$ - $i$  combination with smallest distance is not in path yet) then
    add  $j$ - $i$  combination with smallest distance to matching path;
    add smallest distance to total distance;
  end
end

return  $\frac{TotalDistance}{numberofmatch}$ 

```

et Lempereur [DL11, p. 159] portant sur le design pattern *Modèle-Vue-Contrôleur* et ses différentes variantes.

Afin d'éviter au lecteur une recherche dans le document utilisé pour l'étude de cas, nous rappelons brièvement ici les conclusions trouvées par De Smet et Lempereur. L'étude qui va nous servir est donc *la distance relative entre chemins visuels en fonction du facteur confondant qui est le niveau d'étude des sujets*.

La figure 8.4 présente les moyennes des distances relatives entre chemins visuels en fonction du niveau d'étude et suivant les deux techniques présentées dans cette section. La tendance générale, à savoir que les masters ont une distance inférieure aux doctorants, est respectée par les deux tech-

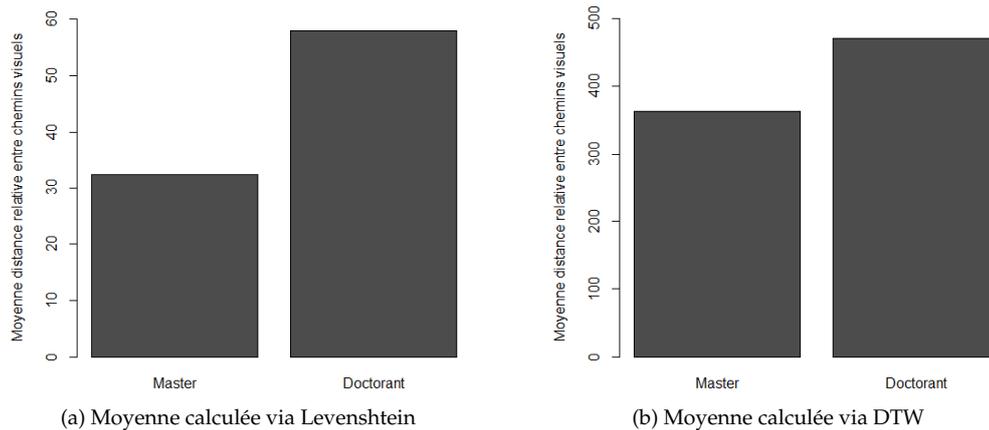


FIGURE 8.4 – Moyenne des distances relatives entre chemins visuels en fonction du niveau d'étude suivant les deux techniques (Levenshtein et DTW)

niques.

Dans cette analyse, les auteurs étaient arrivés à une *p-value* faible et largement inférieure à 0.05 (plus précisément, elle était égale à $2.463e-08$). Avec cette *p-value* et la tendance du graphique de la figure 8.4, ils sont arrivés à la conclusion suivante (avec une probabilité d'erreur inférieure à 5% que :

La Distance Relative entre Chemins Visuels est, en moyenne, plus faible pour les Master que pour les Doctorants.

Dés lors, nous avons réalisé le même calcul avec les mêmes données de départ mais via l'algorithme DTW. Nous sommes arrivés à une *p-value* également inférieure à 0.05 mais pas aussi faible que la valeur calculée par Levenshtein. En effet, notre *p-value* est égale à 0.01922. Cependant, vu qu'elle est inférieure à 0.05 et que le graphique relatif à DTW ressemble à celui de Levenshtein (voir la figure 8.4), nous pouvons déclarer la même conclusion avec une probabilité d'erreur inférieure à 5% que :

La Distance Relative entre Chemins Visuels est, en moyenne, plus faible pour les Master que pour les Doctorants.

Par conséquent, nous pouvons constater que les deux méthodes démontrent les mêmes conclusions, à partir des mêmes données. Les deux techniques présentées peuvent donc être qualifiées d'identiques. Néanmoins, l'approche DTW nous semble la plus appropriée, au vu des avantages que nous avons énoncés dans la section précédente.

Deuxième partie

Experimentation

Définition de l'expérience

La mise en place d'une expérience nécessite une démarche rigoureuse et planifiée. Afin que le lecteur puisse comprendre l'intérêt de celle-ci, il est important de bien définir les éléments qui la constituent. Dans un premier temps, les frameworks, les designs pattern, les représentations, les questions, le matériel utilisé, les métriques ainsi que les sujets retenus pour l'expérience sont présentés dans la section 9.1. Enfin, la section 9.2 explique comment ces différents éléments ont été agencés afin de réaliser notre expérience. Plus concrètement, la planification et l'exécution de l'expérience y sont détaillées.

Sommaire

9.1	Éléments constitutifs de l'expérience	84
9.1.1	Logiciels utilisés comme objet de l'étude	84
9.1.2	Patron de conception	87
9.1.3	Représentation	87
9.1.4	Questionnaire	89
9.1.5	Sujets	91
9.1.6	Matériel	93
9.1.7	Environnement	93
9.1.8	Métriques	94
9.1.9	Zones d'intérêt	96
9.2	Mise en place de l'expérience	99
9.2.1	Définition	99
9.2.2	Planification	99
9.2.3	Exécution	101

9.1 Éléments constitutifs de l'expérience

9.1.1 Logiciels utilisés comme objet de l'étude

Sélection des logiciels

Le premier élément constitutif d'une expérience en génie logiciel représente le support utilisé lors des tâches. Dans notre cas, les tâches de compréhension doivent être réalisées sur des programmes ou des parties de programme, que nous appellerons *frameworks*.

Le choix du nombre de frameworks porte des conséquences sur les résultats obtenus. La menace appelée « mono-opération » expose un biais concernant la généralisation de l'expérience et doit être prise en compte lorsque l'expérience ne considère qu'un seul framework. Ne voulant pas accepter cette menace, nous avons décidé d'utiliser deux frameworks. Dépasser ce nombre peut être dangereux dans une expérience. En effet, un nombre plus élevé implique nécessairement une expérience plus longue pour les sujets. Or, ceux-ci risquent alors de souffrir d'une fatigue physique et psychologique. Enfin, une augmentation de nombre de frameworks demande aussi un nombre plus important de sujets afin d'avoir une répartition équilibrée. Au vu des dernières expériences réalisées au sein de l'École Polytechnique de Montréal, il nous semblait important de ne pas surestimer le nombre de sujets réellement disponibles.

La sélection des frameworks est importante pour les résultats de l'expérimentation. Afin de déterminer ceux-ci et pour éviter, une fois encore, de biaiser les résultats, il est nécessaire de déterminer certaines contraintes qu'ils doivent respecter. Les critères de sélection que nous avons retenus pour notre expérience sont présentés ci dessous :

1. Le framework possède un code ouvert afin de pouvoir faire le diagramme de classes.
2. Le framework n'est pas encore utilisé dans une étude précédente du laboratoire "PtiDej".
3. Le framework contient au minimum un patron de conception de chaque type (créationnel, comportemental ou structurel) (voir section 2.4).
4. Le framework possède un minimum de cinq patrons de conception présents¹.
5. Le framework possède des classes possédant plusieurs rôles dans les patrons de conception.
6. Le framework est largement utilisé par l'industrie.
7. Le framework n'est pas un projet couramment étudié par les sujets potentiels.

De la multitude des frameworks connus, seulement cinq respectaient cette liste pour notre expérience : « *MapperXML (v1.9.7)*, *PMD (v1.8)*, *JRefactory (v2.6.24)*, *Nutch (v0.4)* et *NetBeans (v1.0.x)* ». A partir de cette liste, nous avons examiné chacun des projets *Java* afin de sélectionner les deux meilleurs. Les critères trois et quatre sont les critères qui demandent le plus d'attention du fait qu'ils ont le plus de répercussions. En effet, ils touchent directement aux buts des représentations évaluées au sein de notre expérience. Avoir un framework qui ne contient aucun ou peu de patron n'a aucune valeur à nos yeux.

Plusieurs articles se rapportant à la détection de patrons de conception existent dans la littérature ([TCS+06],[AGJ01],[GSZ04]). Ces documents présentent des techniques de détection de patrons de

1. En essayant de ne pas prendre ceux déjà présents dans l'expérience de Cepeda Porras [Por08].

conception et les programmes qui implémentent ces techniques². Cependant cette étape nécessite beaucoup de temps, le laboratoire PtiDej nous a donc fourni une bibliothèque de logiciels libres pour lesquels les patrons de conception avaient déjà été détectés et validés manuellement. A partir de ce nouveau listing, le nombre de frameworks intéressants a été plus restreint. En fin de compte, les logiciels « **Nutch** » et « **JRefactory** » ont été choisis. Bien que le rôle de ces frameworks en tant que tel n'ont pas beaucoup d'utilité pour la compréhension de notre expérience, nous expliquons brièvement les deux que nous avons sélectionnés afin de donner une idée globale au lecteur :

Apache Nutch est un projet open source de recherche web écrit en Java. Il permet d'indexer des sites Web, intranets et des systèmes de fichier. Il supporte beaucoup de formats de fichiers différents (HTML, PDF, MS Office, OpenOffice, ...) et permet l'analyse de plus de cinquante langues. Nous le nommons Nutch dans le reste de ce document.

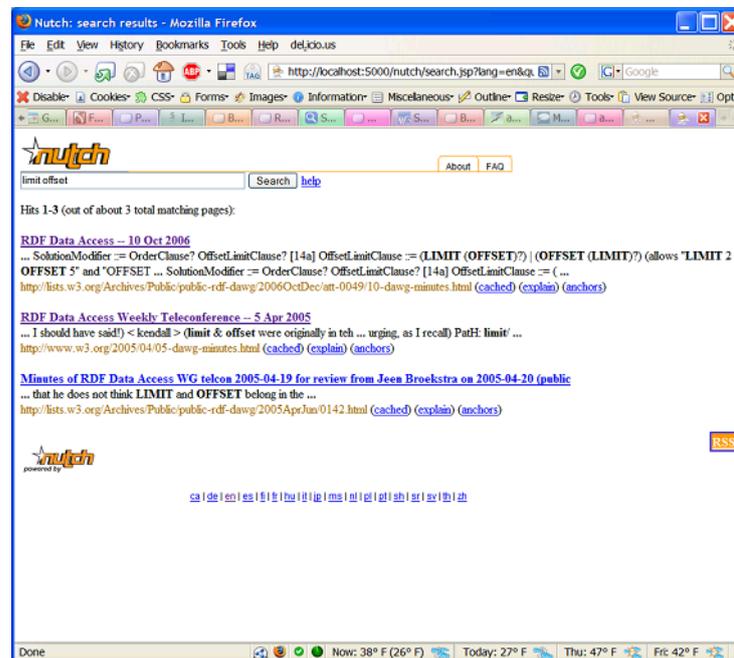


FIGURE 9.1 – Une démonstration d'utilisation de Nutch

JRefactory est un outil de refactoring pour le langage de programmation Java. Il inclut l'imprimante JavaStyle, une visionneuse de diagrammes de classes UML, un vérificateur de standards du code et il calcule des métriques sur le programme. Il peut coopérer avec quelques IDE (NetBeans, JBuilder,...) sous forme de plugin mais il peut aussi être utilisé en ligne de commande.

Langage

Afin de créer les diagrammes de classes des deux frameworks retenus, nous avons décidé de procéder par rétro-ingénierie. Concrètement, les diagrammes sont obtenus par l'application de l'outil « *Visual Paradigme for UML* » sur le code source de ces frameworks. Cette méthode permet de faciliter la création mais aussi de diminuer la probabilité d'un éventuel biais introduit par l'expérimentateur. Cependant, cette technique n'étant pas parfaite, nous sommes repassés sur ces diagrammes afin de les corriger si nécessaire. Lors de cette étape, nous avons vérifié également si les structures

2. <http://java.uom.gr/nikos/pattern-detection.html> et « PtiDej » développé par le laboratoire du même nom.

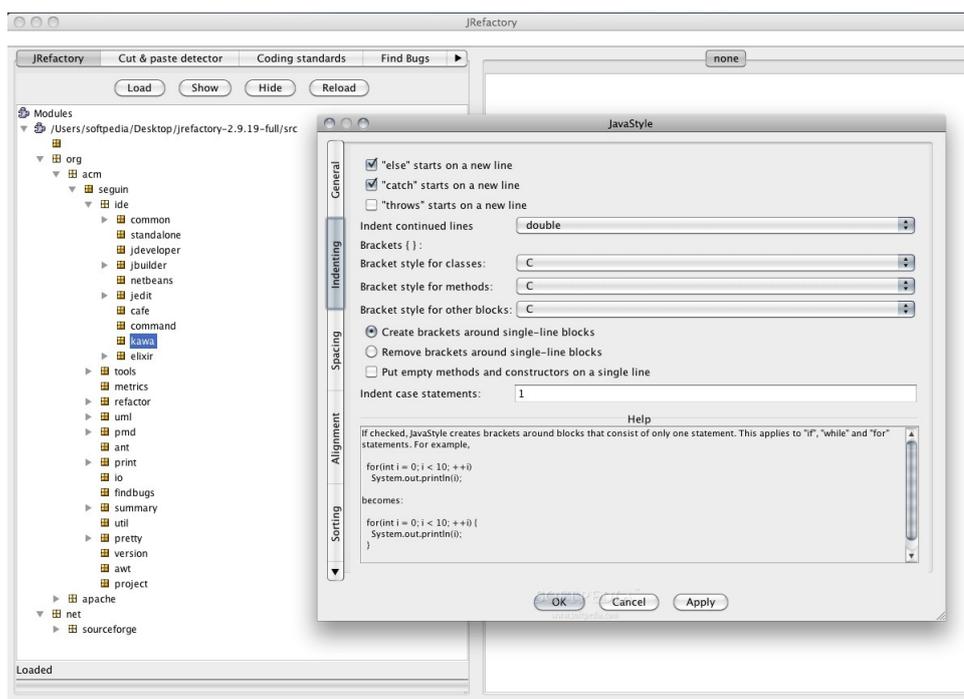


FIGURE 9.2 – L'interface de JRefractory

des patrons de conception présents étaient bien respectés. Pour cette vérification, la référence que représente le livre de Gamma *et al.* [GHJ+94] sur les patrons de conception nous a été très utile.

Similarité des frameworks

Vu la grandeur de ces deux frameworks, seuls les classes intervenants dans les design patterns ont été retenues. De plus, la complexité propre à chacun ne doit pas entrer en compte dans l'expérience, c'est pour cette raison que nos deux frameworks, Nutch et JRefractory, ont été réduits tous les deux à vingt classes.

Si le nombre vingt a été retenu ce n'est pas un hasard. En effet, ce nombre représente la limite de compréhension, ou plus précisément, la facilité de compréhension des sujets vis-à-vis des diagrammes de classes. Une fois dépassé ce nombre, il devient compliqué pour le lecteur lambda de pouvoir bien lire le schéma. Cepeda Porras dans son expérience a comparé plusieurs schémas contenant des diagrammes de tailles différentes. Concrètement, la moitié des diagrammes étaient limités à vingt classes alors que l'autre moitié était composée de diagrammes contenant quarante classes. Si les réponses des sujets qui ont eu affaire aux diagrammes réduits étaient intéressantes et analysables, il en est tout autre des réponses concernant les diagrammes plus conséquents. Cette inégalité empêchera Cepeda Porras de pouvoir établir une conclusion sur les diagrammes de quarante classes, mis à part qu'une complexité trop importante est une barrière non négligeable vis-à-vis de la pertinence des réponses des sujets [Por08].

Afin de ne pas perdre du temps et, surtout, des sujets, l'expérience s'est donc limitée uniquement à des diagrammes de vingt classes.

De plus, certains éléments des classes restantes ont été enlevés. C'est le cas par exemple des accesseurs (les *getters* et les *setters*) qui n'apportaient pas d'informations supplémentaires. Étant donné que les tâches à effectuer n'impliquaient jamais cette notion et par soucis de lisibilité des diagrammes, toutes les cardinalités ont été masquées. Il en va de même pour les rôles des relations. Enfin, vu leur

importance dans les questions, les noms des classes ont été mis en gras.

Néanmoins, malgré le fait que nous avons essayé de rendre les deux frameworks égaux, le lecteur s'apercevra sans mal que le framework Nutch est plus *chargé* que JRefactory. Dû à cette différence et afin de comprendre dans quelle mesure les frameworks influencent les résultats des représentations, les sections 10.2.1, 11.1.1 et 12.1.1 s'intéressent aux résultats des frameworks dans les différentes tâches (respectivement : « composition », « participation » et « rôle »).

9.1.2 Patron de conception

Une fois les frameworks de support choisis, nous nous sommes intéressés aux patrons de conception proprement dits. Si Nutch et JRefactory contiennent un ensemble non négligeable de patrons architecturaux, il faut encore qu'on y trouve au moins un patron créationnel, un patron comportemental et un patron structurel. En effet, afin de respecter la troisième contrainte présentée dans la section 9.1.1, il est important que chacun des frameworks possède un patron de chaque type.

Concrètement, JRefactory contient les design patterns « Adapter », « Builder », « Factory Method », « Singleton », « State » et « Visitor ». Tandis que le framework Nutch se retrouve avec les design patterns « Singleton », « Bridge », « Command », « Memento », « Template Method », « Strategy » et « Visitor ». On peut remarquer facilement que leur nombre total est sensiblement pareil d'un framework à l'autre. En effet, pour garder la même complexité, il était nécessaire de ne pas avoir une grande différence sur le nombre des design patterns. En ce qui concerne la vérification de la présence de chacun des types, elle se fait grâce au tableau 9.1 réalisé à partir des informations provenant de l'ouvrage de Gamma *et al.* [GHJ+94]. Dès lors, le lecteur remarquera directement que Nutch a un patron créationnel, deux patrons comportementaux et cinq patrons structurels tandis que JRefactory contient trois patrons créationnels, un patron comportemental et deux patrons structurels.

Créationnel	Comportemental	Structurel
Abstract Factory	Chain of responsibility	Adapter
Builder	Command	Bridge
Factory Method	Interpreter	Composite
Prototype	Iterator	Decorator
Singleton	Mediator	Facade
	Memento	Flyweight
	Observer	Proxy
	Strategy	
	Template Method	
	Visitor	

Tableau 9.1 – Classification des patrons de conception selon Gamma *et al.* [GHJ+94]

9.1.3 Représentation

Sélection des représentations

Une fois « la version de base » du diagramme de classes obtenue pour chacun des frameworks, nous nous sommes intéressés aux différentes représentations possibles d'un pattern. Comme nous l'avons vu dans la section 3.2, quatre représentations sont susceptibles d'être pertinentes pour notre expérimentation :

- La représentation standard, nommée « UML Collaboration Notation » (appelée *UMLNotation*) [Vli98]
- La représentation « stereotype-enhanced UML diagrams » de Dong *et al.* (appelée *Dong*) [DYZ07]
- La représentation « Pattern-enhanced class diagrams » de Schauer et Keller (appelée *Schauer*) [SK98]
- La représentation « « pattern : role » notation » de Gamma *et al.* (appelée *Gamma*) [Gam96]
- La représentation « Stereotype Pattern Enhanced » créée par nos soins (voir section 3.2.5) (appelée *SPE*)

Cependant, après une analyse de l'étude réalisée par Porras[Por08], nous avons décidé d'écarter certaines représentations. En effet, vu leur faible taux de réussite aux différentes questions dans l'expérimentation, les représentations « « pattern : role » notation » et « Pattern-enhanced class diagrams » n'ont pas été reprises. Concrètement, les représentations « stereotype-enhanced UML diagrams » et « UML Collaboration Notation » ressortent de l'expérience de Cepeda Porras comme étant supérieures en terme d'efficacité. Dans l'optique de ne pas refaire la même expérience mais bien de comprendre pourquoi ces représentations étaient meilleures, nous nous sommes concentrés sur ces deux dernières. En plus des deux représentations choisies, nous avons décidé d'en créer une inédite. L'idée d'utiliser des couleurs comme pour la représentation de Gamma nous semblait pertinente. C'est pour cette raison qu'une nouvelle représentation reprenant cette idée à été créée. La section 3.2.5 explique d'une manière plus détaillée les choix que nous avons faits.

Les sujets ne connaissant pas les trois représentations, il a été nécessaire de produire un mini tutoriel pour chaque technique employée. Ceux-ci seront lus par chaque utilisateur avant le début de l'expérience. Durant cette phase, nous restons à la disposition des sujets. Il est capital que le sujet comprenne tout à fait les représentations avant le début du questionnaire. De plus, il va de soi que nous ne pouvons plus intervenir pour aider le sujet lors des différentes questions. Chaque tutoriel se termine par une question de compréhension qui ressemble aux questions que le sujet trouvera durant l'expérience afin de l'obliger à utiliser la théorie qu'il vient de lire. Si nous prêtons tant d'attention à la compréhension des particularités des différentes représentations, c'est parce qu'elle est capitale vis-à-vis des résultats. Une mauvaise compréhension pourrait influencer négativement les résultats d'une représentation.

Langage

Le choix du langage de modélisation graphique utilisé pour représenter les frameworks dans les différentes représentations s'est imposé de lui-même. La principale raison est que toutes ces représentations utilisent comme base un même diagramme de classes UML. Sans surprise, ce choix du langage de modélisation est identique à l'expérience réalisée par Cepeda Porras [Por08]. De plus, son utilisation fortement répandue en fait un langage plus qu'intéressant. En effet, une exigence de notre expérience impose aux différents sujets de connaître un minimum le support de l'expérimentation. Avec le langage UML, moins de sujets seront exclus de celle-ci.

Similarité des représentations

Pour des raisons pratiques, la disposition des classes ainsi que le nombre de classes et de méthodes sont invariables d'une représentation à l'autre. Enfin, l'organisation graphique de ces représentations a été ajustée dans le but de réduire la différence visuelle des trois différentes représentations d'un même framework.

Les représentations Dong et SPE ont une approche qui est sensiblement pareille. UMLNotation par l'ajout de ses éclipses portant le nom des design pattern, s'éloigne un peu plus des deux premières représentations. Cependant, vu que le but de l'expérimentation est de comparer ces différentes représentations, il n'est pas pertinent de rechercher plus en avant une similarité entre elles.

9.1.4 Questionnaire

Grâce aux supports déterminés dans la section 9.1.3, nous avons pu rédiger les différentes questions qui ont été posées à nos sujets. Ces questions sont nommés des diagrammes.

Avant de présenter les questions au lecteur, il est important de définir comment elles ont été catégorisées. En effet, chaque question se rapporte à une tâche bien précise. Dans notre expérimentation, le sujet doit faire face à trois différentes tâches. Les types de tâche ont été directement repris de l'expérience de Perras [Por08]. En effet, ces types couvrent l'ensemble des questions que l'on peut poser sur la présence de design patterns dans un diagramme de classe. Il est vrai qu'un lecteur possédant une connaissance non nulle des patrons de conceptions pourrait voir en ces questions une certaine facilité. De l'avis de Yann Guéhéneuc et de l'analyse des travaux réalisés par Lempereur et De Smet [DL11], complexifier les questions ne rendent pas l'expérience plus intéressante, que du contraire. D'ailleurs, une limite de leurs travaux avait été la surestimation des connaissances des sujets qui ont, malheureusement, conduit à peu de résultats significatifs. Dans tous les cas, les tâches choisies pour cette expérience portent sur le repérage des informations relatives aux patrons de conception.

Concrètement, le but principal de ces tâches est de mettre les sujets dans des situations à partir desquelles nous pouvons analyser comment ceux-ci utilisent les différentes notations contenues dans chaque représentation. En effet, ils doivent être capables de repérer des informations relatives aux patrons de conception. Une fois cette analyse réalisée, il ne reste plus qu'à l'interpréter correctement afin de comparer le niveau de la charge mentale de chaque sujet.

Nos trois tâches sont donc :

- Détecter les classes appartenant à un design pattern que l'on nomme la tâche « participation ».
- Détecter les design patterns dans lesquels une classe intervient que l'on nomme la tâche « composition ».
- Déterminer les différents rôles joués par une classe que l'on nomme la tâche « rôle ».

Nous avons hésité longuement entre deux ou trois questions pour chacune des tâches. Avoir plus de questions permet de mettre plus facilement en avant les avantages et les inconvénients d'une représentation suivant la tâche donnée. Cependant, le nombre de frameworks étant déjà à deux, l'expérimentation pouvait vite devenir assez longue. Dans le cas où le sujet devrait répondre à l'ensemble des questions des deux frameworks, le nombre de questions attribuées à chaque sujet se verrait fortement augmenté. Afin de ne pas courir le risque de voir les dernières questions bâclées, nous avons décidé de fixer le nombre de questions par tâche à deux. Ces questions ont été choisies

afin que les informations présentes dans les diagrammes soient suffisantes pour trouver la réponse en deux minutes.

Le lecteur trouvera en annexe B les différentes questions. Le tableau 9.2 et le tableau 9.3 montrent les distributions des questions respectivement pour JRefactory et Nutch.

Une particularité de notre expérience réside dans le fait que nous avons divisé chaque question en deux phases distinctes.

La première phase, que l'on nomme *la recherche d'informations* appelée « *question* », consiste à ce que le sujet trouve sur le diagramme l'emplacement des informations qu'il juge pertinente pour répondre à la question. Il ne doit pas les retenir par cœur mais juste savoir où elles se situent.

Lors de la deuxième phase, *le rassemblement d'informations* appelé « *support* », le sujet est amené à noter sa réponse sur papier. Pour se faire, il peut bien évidemment s'aider du diagramme. Cependant, le diagramme ne doit pas servir d'outil de recherche mais bien de support.

JREFACTORY		
Q1	Composition	Identifiez tous les design patterns dans lesquels participe la classe UMLNode
Q2	Composition	Identifiez tous les design patterns dans lesquels participe la classe AddMethodTypeVisitor
Q3	Participation	Identifiez toutes les classes qui participent aux deux instances du design pattern Builder
Q4	Participation	Identifiez toutes les classes qui participent au design pattern Visitor
Q5	Rôle	Identifiez tous les rôles que joue la classe FileSummary
Q6	Rôle	Identifiez tous les rôles que joue la classe UMLNodeViewerFactory

Tableau 9.2 – Les six questions relatives au framework JRefactory

NUTCH		
Q1	Composition	Identifiez tous les design patterns dans lesquels participe la classe Page
Q2	Composition	Identifiez tous les design patterns dans lesquels participe la classe FetchListTool
Q3	Participation	Identifiez toutes les classes qui participent au design pattern Command
Q4	Participation	Identifiez toutes les classes qui participent au design pattern Iterator
Q5	Rôle	Identifiez tous les rôles que joue la classe WebDBWriter
Q6	Rôle	Identifiez tous les rôles que joue la classe PagesByUrlProcessor

Tableau 9.3 – Les six questions relatives au framework Nutch

9.1.5 Sujets

Sélection des sujets

Les sujets ayant passé l'expérience sont majoritairement issus du laboratoire Ptidej³ et du Soccer Lab⁴ et plus généralement du département de Génie Logiciel de l'École Polytechnique de Montréal. Quatre étudiants des FUNDP et plusieurs chercheurs de l'université de Montréal ont également participé à l'expérience.

La démarche d'invitation des sujets a été réalisée par mail, à partir des adresses fournies par le directeur des équipes de recherche. Un doodle a été mis en place afin de permettre aux différentes personnes de choisir leur plage horaire la plus appropriée.

Cependant, comme pour l'étude réalisée l'année dernière par Lempereur et De Smet [DL11], peu de personnes ont montré un réel engouement pour notre étude. Afin de motiver plus de gens, nous avons décidé d'offrir une boisson aux sujets et de venir dans les laboratoires afin de leur en parler de vive voix. Étant donné les critères préliminaires, le temps nécessaire pour réaliser l'expérience avec un sujet et le temps qui nous était imparti pour faire passer un certain nombre de sujets, nous n'avons pas jugé nécessaire de faire un appel public de type affiches ou annonces. Cependant, sachant que le nombre de sujets est déterminant pour des expériences telles que la nôtre, nous avons décidé de faire passer un *sujet-test* avant de contacter véritablement les membres des laboratoires. Par ce test, nous avons pu vérifier l'apparition ou non de problème majeur lié au matériel lors de son passage. Ainsi, nous avons pu écarter le risque de devoir annuler les résultats de plusieurs personnes suite à un dysfonctionnement repéré bien plus tard.

Chaque sujet est amené à répondre à deux séries de questions propre à un framework (comme expliqué en section 9.1.4). Avant le début de l'expérience, le sujet reçoit un tutoriel des différentes représentations qu'il devra utiliser. Au vu du caractère cosmopolite des différents sujets, nous avons réalisé ce questionnaire dans les deux langues couramment parlées dans le laboratoire, c'est-à-dire le français et l'anglais (certains sujets étaient unilingue). À noter que les représentations utilisées par les sujets suivent un roulement. Ainsi, le premier sujet n'aura pas les mêmes représentations que le deuxième sujet et ainsi de suite... Pour plus de détails, le lecteur peut se rendre à la section 9.2 qui est entièrement dédiée, comme son nom l'indique, à la mise en œuvre de l'expérience.

Pour chaque sujet potentiel, nous nous assurons que celui-ci répond bien à quelques critères préliminaires. Premièrement, le sujet ne doit pas avoir de problèmes oculaires graves empêchant le fonctionnement de l'oculomètre. Deuxièmement, seuls les sujets étant familiarisés avec le paradigme orienté objet sont retenus. Et enfin, ceux-ci doivent avoir une bonne connaissance d'UML.

Afin d'évaluer le niveau de connaissance en UML et en patron de conception des différents sujets ayant pris part à notre expérience, nous avons mis en place un post-questionnaire, que chaque sujet a dû remplir à la fin de la séance d'oculométrie. De plus, nous profitons également de ce questionnaire pour récolter des informations pouvant nous aider à expliquer le rôle des facteurs confondants (sexe, niveau d'étude). Au vu de l'importance de ce questionnaire, celui-ci a été rendu obligatoire pour l'ensemble des sujets. Cependant, comme pour le reste de l'expérience, il restait totalement anonyme. Le sujet ne devait s'identifier que par le numéro reçu avant de passer l'expérience *réelle*. En aucun cas, ces numéros n'étaient associés à un nom ou un prénom. Vu l'obligation de remplir ce questionnaire, et pour les mêmes raisons citées pour le questionnaire principal, nous avons réalisé

3. <http://www.ptidej.net>

4. <http://web.soccerlab.polymtl.ca>

deux post-questionnaires, un en français, l'autre en anglais.

Grâce à ce post-questionnaire, les variables suivantes sont connues pour chaque sujet :

- Sexe : le sexe du sujet (Homme - Femme).
- Niveau d'étude le plus haut : le niveau d'étude du sujet (Bac – Master – Doctorat - Autre).
- Niveau de connaissance UML : le niveau de connaissance en UML du sujet (dans l'intervalle [0..2]).
- Niveau de connaissance en patrons de conception : le niveau de connaissance en patrons de conception (dans l'intervalle [0..1]). Ces questions couvrent les concepts généraux jusqu'aux structures plus spécifiques.
- Expérience en maintenance : si le sujet a déjà effectué des tâches de maintenance sur des diagrammes de plus de 20 classes (1) ou non (0).
- Expert : si le sujet est considéré comme un expert (1) ou non (0). Cette valeur dépend du niveau de connaissance UML, en patrons architecturaux et selon son expérience de maintenance.

Afin d'être le plus utile, ces différentes variables sont retranscrites dans les fichiers d'input de TAUPE. En effet, chaque sujet se voit associer un fichier « *.subject* » où sont retranscrits ces informations⁵ qui seront utilisées plus tard pour les analyses.

Pour le niveau de connaissance UML et le niveau de connaissance des patrons de conception, le questionnaire propose des questions de plus en plus difficiles. Cette approche évite de perdre le sujet dans trop de détails. En effet, en plaçant des questions plus générales au début, le sujet se remémore ses connaissances et peut, ensuite, s'intéresser aux détails. De plus, en attribuant un ratio calculé suivant la complexité de la question, il est facile d'obtenir un score final représentant réellement la connaissance du sujet.

Nous décidons également d'évaluer la charge mentale des sujets grâce au formulaire subjectif « NASA-TLX » précédemment utilisé par Lorent LEMPEREUR et Benoit DE SMET [DL11]. Une fois encore, dans un souci de clarté pour les futurs utilisateurs, nous rédigeons l'entièreté de ce documents en français et en anglais.

Rejet

Malheureusement, malgré nos précautions, quelques sujets ont dû être retiré de l'expérience. Concrètement, sur vingt sujets passés, trois participants ont été *supprimés* de notre expérimentation. Le taux de rejet pour notre expérience est donc de quinze pour cent⁶.

Sur les trois sujets inexploitable, deux sujets ont été éliminés à cause de problème technique. Pour un des deux, le logiciel FaceLAB a *buggé* lorsqu'il répondait aux questions. Ce bug a rendu impossible la récupération des données oculaires. Dès lors, nous avons été obligé de le retirer. Le deuxième sujet retiré est dû à un problème qui ne concerne qu'une seule question. En effet, toutes ses données s'étaient bien enregistrées excepté pour une question de l'expérience. Malgré le fait qu'il ne s'agissait que d'un problème très local, nous avons décidé de le retirer afin de ne pas biaiser les résultats de cette question. La source de l'élimination du dernier sujet écarté n'est pas un problème technique mais un mauvais placement du sujet. En effet, comme expliqué dans la section 7.1, l'eye-tracker demande une attention particulière par rapport au placement du sujet, d'autant plus si celui-ci porte

5. Malheureusement, ces informations sont encore retranscrites manuellement.

6. Ce qui est un bon résultat comparé à l'expérience réalisée l'année dernière où le taux de mortalité était de 28,6%.

des lunettes. Or ici, le sujet éliminé portait des lunettes et n'était pas idéalement placé. Ce mauvais placement a entraîné un faible taux d'enregistrement de ses mouvements oculaires et une augmentation du nombre d'offsets.

9.1.6 Matériel

Matériel d'enregistrement

Comme expliqué dans la section 7.1, l'École Polytechnique de Montréal possède des équipements dans le domaine de l'oculométrie. Au vue de cet équipement, nous nous sommes intéressé aux études réalisées avec celui-ci. Comme attendu, le laboratoire PtiDej est spécialisé dans ce domaine. De plus, comme nous l'avons expliqué dans la section 5.2.2, les *mesures physiologiques* (qui incluent les mouvements oculaires) sont pertinentes afin d'analyser la charge mentale d'un sujet. L'expérience nécessite de faire un choix concernant le support d'enregistrement à utiliser. Notre décision se porta, presque instinctivement, sur FaceLAB. En effet, bien que l'Eye-LinkII soit un matériel puissant, FaceLAB présente bien plus d'avantages. Parmi ceux-ci, on retiendra le fait que ce système est plus récent, plus précis, moins intrusif et plus facile d'utilisation. Notre expérience demandant un laps de temps non nul, il est important de laisser au sujet une liberté de mouvement (et ce notamment au niveau de la tête). C'est pourquoi il est essentiel de favoriser un oculomètre non intrusif.

Matériel d'interrogation

En ce qui concerne le support d'interrogation, nous nous sommes dirigés vers une version papier. En effet, FaceLAB et GazeTracker ne permettent pas d'intégrer dans l'expérience une page où le sujet pourrait taper sa réponse à l'écran. L'utilisation d'un support papier amène avec lui un lot de problèmes potentiels. Le sujet peut, par exemple, prendre le support en main et cacher ses yeux des caméras. En effet, le fait d'utiliser un système aussi peu intrusif peut parfois faire oublier aux sujets que leurs mouvements des yeux sont enregistrés. Afin d'éviter toute mauvaise surprise, un rappel au début de chaque expérience est fait. De plus, une personne s'occupe durant toute la durée de l'expérimentation de vérifier si les caméras captent toujours bien les pupilles du sujet. Néanmoins le support papier possède un avantage considérable. Celui-ci autorise le sujet à prendre connaissance de la question avant l'apparition du diagramme. Cependant, la question n'était pas exclusivement sur le support papier, elle était également répétée en haut à gauche du diagramme afin que le sujet puisse retrouver le nom de la classe/du design pattern recherché sans quitter l'écran des yeux.

9.1.7 Environnement

L'environnement que nous avons choisi a été grandement inspiré de celui mis en place par Lempereur et De Smet[DL11]. En effet, cet environnement présente l'avantage de laisser le sujet libre tout en permettant de le surveiller via un contrôleur. La figure 9.3 représente cet environnement :

- A : Cette zone représente la partie réservée au sujet. Il peut disposer ainsi de sa feuille et répondre aux questions d'une manière autonome.
- B : Ecran où étaient affichés les diagrammes. Cet écran est en fait relié à l'ordinateur portant le tag F.
- C : Caméras de FaceLAB. Il est important que le sujet ne place rien entre elles et lui.

D : Clavier permettant au sujet de se « déplacer » dans l'expérience.

E : Cette zone représente la partie réservée à l'examineur. On voit clairement que celle-ci est bien délimitée par une paroi de la zone du sujet (portant le tag A).

F : Ordinateur central. L'écran (tag B), le clavier (tag D) ainsi que les caméras de FaceLAB (tag C) sont reliés à cet ordinateur. Il est utilisé pour enregistrer les données de l'oculomètre et aussi pour surveiller, via les outils de GazeTracker, la fiabilité des enregistrements.

G : Chaise de dentiste où le sujet s'installe. Malgré son côté *effrayant*, cette chaise a l'avantage d'obliger les sujets à ne pas trop bouger.

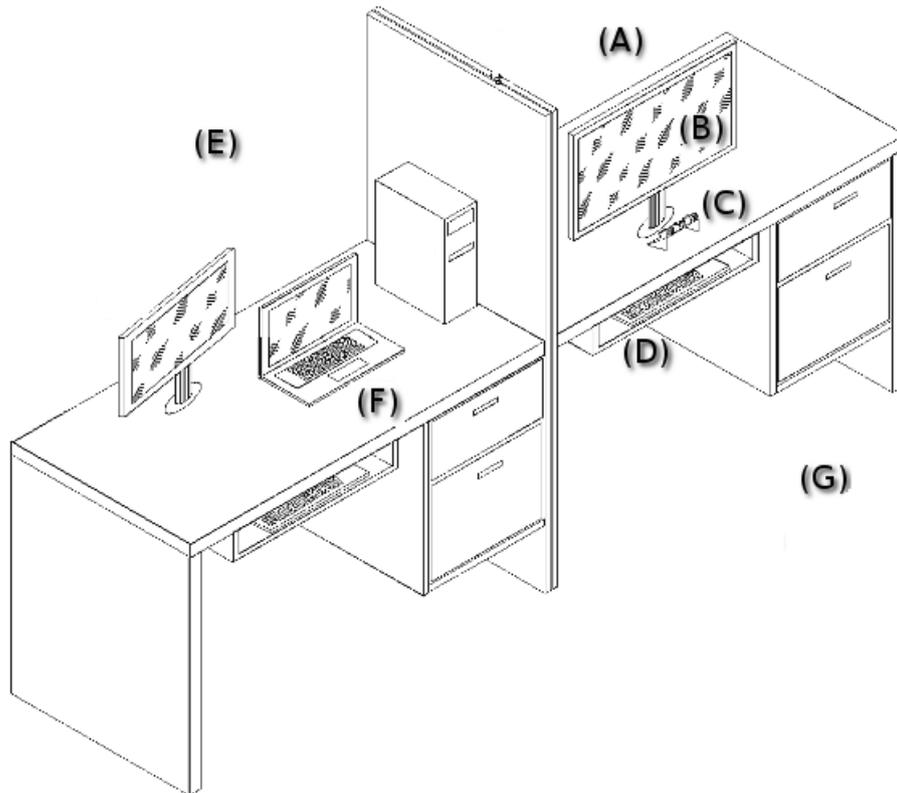


FIGURE 9.3 – Environnement de l'expérimentation.

9.1.8 Métriques

Vu que notre expérience se résume à des données récoltées via un oculomètre, nous nous sommes intéressés à l'ensemble des métriques existantes dans ce domaine (présentées dans la section 4.3.3). Cette section se divise en deux parties. En effet, vu que chaque question possède une phase « question » et une phase « support » (voir section 9.1.4), les métriques ont également été adaptées à ces deux phases.

Métriques utilisées pour la recherche d'informations

Lors de la phase de recherche d'informations, le sujet est amené à examiner le diagramme afin de trouver les éléments qui serviront à répondre à la question. Lors de cette recherche, les fixations ainsi que les saccades peuvent donner des informations sur la technique de recherche suivant les diagrammes. L'ensemble des métriques de cette liste ont été utilisées pour analyser ces techniques de recherche :

- *Validité des réponses* : Cette métrique indique si la représentation utilisée est utile ou inutile pour le sujet. Un pourcentage de réponse valide assez haut révèle que le sujet a réussi à mettre les informations pertinentes ensemble afin de trouver la bonne réponse, et donc, que la représentation l'a bien aidée dans cette tâche.
- *Temps passé sur une tâche* : Via le temps passé sur une tâche, on peut savoir dans quelle mesure la représentation présente les données pertinentes. En effet, si le temps est sensiblement plus grand pour une représentation donnée, cela signifie que les informations utiles ne sont pas idéalement bien placées.
- *Densité spatiale* : Cette métrique donne des informations sur la recherche d'informations. Une valeur petite indique que les informations pertinentes ont été détectées rapidement. Tandis qu'une valeur élevée traduit une recherche laborieuse pour le sujet.
- *Densité de transition* : Une densité de transition faible indique des transitions directes, alors qu'une densité plus élevée révèle un va-et-vient continu et donc une incertitude de la part du sujet.
- *Pourcentage de transition AORI vers AOII* : Cette métrique nous donne un information sur l'importance que donne le sujet aux informations pertinentes du diagramme. Un pourcentage de transition AORI vers AOII élevé indique une mauvaise interprétation de ces données pertinentes.
- *IN AORI / IN AOII normalisé (par zone d'intérêt et par surface)* : Via cette métrique, on peut voir si l'effort prodigué par le sujet dans cette tâche de recherche d'informations est pertinent ou pas. Un résultat élevé correspond à un sujet réalisant peu d'effort et un taux faible correspond à un sujet réalisant beaucoup d'effort.
- *Ratio de la durée des fixations et des saccades* : Pour cette métrique, la durée des saccades est associée à la recherche et la durée des fixations au traitement d'informations. Cette métrique exprime la proportion du temps de traitement vis-à-vis du temps de la recherche.
- *Moyenne des distances relatives des chemins visuels* : Le but de cette métrique est de calculer les moyennes des distances relatives des chemins visuels de différents groupes de sujets qui sont différenciés par des critères (sexe, études, connaissance du domaine de l'expérimentation,...). Ainsi, on peut voir si des comportements ressortent parmi ces différents groupes.
- *Durée moyenne des fixations* : Cette durée est un excellent moyen pour savoir si le sujet doit s'arrêter longtemps ou pas sur les données pour comprendre leur intérêt.
- *Nombre moyen de fixations Post-AORI* : Cette métrique calcule la moyenne du nombre de fixations que le sujet a continué à faire après la visualisation d'une zone contenant des informations pertinentes. Si le résultat de cette métrique est grande, cela se traduit par une difficulté réelle du sujet de lire et comprendre l'information pertinente. A l'opposé, un résultat proche de zéro indique que les données pertinentes sont facilement détectables.

Métriques utilisées pour le rassemblement d'informations

Pour le rassemblement d'informations, le choix se porte vers des métriques qui n'utilisent pas les saccades. Ce choix se justifie par le fait que le sujet note sa réponse lors de cet acte. Ainsi, il se peut que l'oculomètre enregistre des fausses saccades. Des métriques utilisant ces saccades donneraient forcément des réponses biaisées. Les métriques retenues sont les suivantes :

- *Densité spatiale* : Cette métrique donne des informations sur l'efficacité du sujet à rassembler les données pertinentes. Plus le résultat de cette métrique est petite, plus la recherche est dirigée. En d'autres termes, que le sujet savait où rechercher les données. Au contraire, une valeur élevée indique que les informations sont dispersées sur l'ensemble du diagramme.

- *IN AORI / IN AOII normalisé (par zone d'intérêt et par surface)* : Comme pour la tâche de recherche d'informations, cette métrique indique ici si l'effort du sujet dans la tâche de rassemblement d'éléments est pertinent ou pas.
- *Durée moyenne des fixations* : La durée moyenne des fixations est une bonne indication quant à la concentration que demande le diagramme pour que le sujet puisse lire et utiliser les données utiles pour répondre à la question.

9.1.9 Zones d'intérêt

L'apparition des métriques nous amène à un autre problème. Quelles sont les zones d'intérêt et quelles sont les zones de non-intérêt ? Il n'y a évidemment pas de réponse générique. Les zones d'intérêt dépendent toujours de la question demandée et du diagramme présenté.

Nous avons donc identifié nous mêmes les zones d'intérêt des trente-six diagrammes créés⁷. Nous avons considéré comme zone d'intérêt, tout élément graphique (classe, patron⁸ et légende) qui doit être vu afin de répondre à la question. S'il existe plusieurs façons de répondre à la question, tout les éléments qui appartiennent à une façon de répondre sont considérés comme une zone d'intérêt.

Une zone neutre a été spécialement définie à l'endroit où se trouve la question, en haut à gauche. Bien que celle-ci soit utile pour le sujet, elle ne peut être considérée comme une zone d'intérêt.

Les tableaux 9.4 et 9.5 suivants présentent les différentes zones d'intérêt pertinentes et non pertinentes en fonction des éléments graphiques des représentations. Les colonnes contiennent les six questions, et pour chacune d'entre elles, les trois représentations sont différenciées. La lettre « U » est utilisée pour la représentation UMLNotation, la lettre « D » pour la représentation de Dong et la lettre « S » pour la représentation SPE. Si ce numéro est égal à 1, cela signifie que l'élément est une zone d'intérêt. Le numéro 0 signifie le contraire. Comme nous l'avons expliqué dans cette section, les zones où sont placées les questions ont un intérêt nul (représenté par une barre). A noter que les numéros des questions se rapportent à la définition donnée en section 9.1.4. Comme expliqué précédemment, les zones d'intérêt ne considèrent que les éléments du diagramme qui apportent une information. Les tableaux reprennent ces différents éléments en les classant. Bien sûr, certaines représentations ne reprennent pas l'ensemble de ces éléments dûs à leur configuration. Lorsque l'élément n'apparaît pas dans un diagramme, une cellule noire est présente dans le tableau.

7. Il y a six questions par représentation par framework. Le nombre de frameworks étant de deux, et le nombre de représentation de trois, le nombre total de diagrammes est trente-six.

8. Cet élément est en fait l'ellipse où figure le nom du patron de conception présent dans la représentation UMLNotation.

Question	Q1			Q2			Q3			Q4			Q5			Q6		
	U	D	S	U	D	S	U	D	S	U	D	S	U	D	S	U	D	S
CLASSES																		
EMBuilder	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Node	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
ExtractMethodRefactoring	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
MenuBuilder	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
UMLNode	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
BrowerAdapter	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NewProjectAdapter	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MultipleDirClassDiagramReloader	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NodeViewerFactory	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UMLNodeViewerFactory	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
UMLNodeViewer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NodeViewer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SummaryLoaderState	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FileSummary	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
Summary	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
SummaryVisitor	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
TraversalVisitor	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
PrintVisitor	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
AddMethodTypeVisitor	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0
NearMissVisitor	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
PATRONS																		
Builder[1]	0			0			1			0			0			0		
Builder[2]	1			0			1			0			0			0		
Adapter	1			0			0			0			0			0		
Factory Method	0			0			0			0			0			0		
Singleton	0			0			0			0			0			0		
State	0			0			0			0			0			0		
Visitor	0			1			0			1			0			0		
LEGENDE																		
Légende				1			1			1			1			1		

Tableau 9.4 – Attribution des zones d'intérêt de chaque question pour le framework JRefractory

Question	Q1			Q2			Q3			Q4			Q5			Q6		
	U	D	S	U	D	S	U	D	S	U	D	S	U	D	S	U	D	S
PATRONS																		
EnumCall	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PageEnumCall	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NullWritable	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WebDBWriter	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
IWebDBWriter	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
KeyExtractor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PageURLExtractor	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Page	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Link	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
WritableComparable	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PagesByURLProcessor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
LinksByMD5Processor	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Writer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CloseProcessor	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
DistributerWebDBWriter	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
WebDBReader	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
DistributedWebDBReader	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
FetchListTool	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
IWebDBReader	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
DistributedAnalysisTool	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
ASSOCIATION																		
Template Method	0			0			0			0			0			0		
Singleton	0			0			0			0			0			0		
Bridge	1			0			0			0			0			0		
Command	1			0			1			0			0			0		
Iterator	0			0			0			1			0			0		
Strategy	0			1			0			0			0			0		
Memento	0			0			0			0			0			0		
LEGENDE																		
Légende			1			1			1			1			1			1

Tableau 9.5 – Attribution des zones d'intérêt de chaque question pour le framework Nutch

9.2 Mise en place de l'expérience

Cette section présente respectivement la définition de l'expérience, sa planification et son exécution selon les termes de Wohlin *et al.* [WRH+00] présentés dans la section 6.

9.2.1 Définition

La définition de l'expérience peut être décrite telle que :

Analyse des représentations visuelles « stereotype-enhanced UML diagrams » de Dong *et al.*, « UML Collaboration Notation » et « Stereotype Pattern Enhanced ».

Dans le but d' évaluer l'impact des différentes représentations.

Vis-à-vis de leur : visualisation des designs patterns dans un diagramme de classe UML.

Du point de vue des développeurs

Dans le contexte d' étudiants de Master et de Doctorat du Département de Génie Logiciel de l'École Polytechnique de Montréal.

Objet d'étude Les représentations sont utilisées pour afficher des diagrammes de classe UML de parties de projets réels. Ces diagrammes sont décrits en section 9.1.4.

But L'objectif de l'expérience est la comparaison de ces représentations les unes avec les autres.

Focalisation de la qualité L'impact étudié est l'effet qu'ont ces représentations sur la compréhension de programmes comme définie en section 3.

Perspective La compréhension de programme est considérée du point de vue des développeurs et plus particulièrement des mainteneurs de logiciels.

Contexte Les sujets sont des étudiants en Génie Logiciel dans le cadre du Master ou d'un Doctorat à l'École Polytechnique de Montréal, de la Faculté d'Informatique de Namur ou du département d'Informatique et de Recherche Opérationnelle (DIRO) de l'Université de Montréal. L'environnement est un laboratoire de l'École Polytechnique de Montréal dédié au déroulement de l'expérience.

9.2.2 Planification

La plupart des éléments qui constituent la planification de l'expérience sont détaillés au début de ce chapitre en section 9.1. Cette section est structurée selon la planification de l'expérience décrite par Wohlin *et al.* [WRH+00].

Formulation des hypothèses

Notre approche emploie ces trois types d'analyses : l'analyse de la performance, l'analyse des mesures physiologiques et l'analyse des mesures subjectives. Nous allons évaluer les différentes hypothèses nulles pour chacune des tâches choisies (« composition », « participation » et « rôle »). Les chapitres 10, 11 et 12 détaillent plus en avant ces hypothèses.

Variables indépendantes

Suivant les hypothèses, nous avons 2 variables indépendantes :

- Frameworks : deux traitements (JRefactory et Nutch) (cf. section 9.1.1)
- Représentations : trois traitements (UMLNotation, Dong et SPE), tous avec la même densité (cf. section 9.1.3)

Variables dépendantes

Les variables dépendantes sont choisies en fonction de nos hypothèses, des variables indépendantes et des capacités du système d'eye-tracking.

L'analyse de la performance comprend deux variables dépendantes :

1. Pourcentage de réponses correctes
2. Temps nécessaire aux réponses

L'analyse physiologique possède huit variables dépendantes :

1. Densité spatiale
2. Densité de transition
3. Pourcentage de transition AORI vers AOII
4. IN AORI/ IN AOII normalisé
5. Ratio de la durée des fixations et des saccades
6. Moyenne distance relative entre chemin visuel (DTW)
7. Durée moyenne des fixations
8. Nombre moyen de fixations Post-AORI

Enfin, l'analyse des mesures subjectives ne reprend qu'une seule variable dépendante :

1. Méthode NASA-TLX pour l'évaluation de la charge mentale

Variables de l'atténuation ou facteurs confondants

Nous avons identifié deux facteurs confondants :

- Sexe (femme/homme)
- Niveau d'étude (Master/Doctorant)

Sélection des sujets

La section 9.1.5 décrit la sélection des sujets.

Conception de l'expérience

Nous avons deux variables indépendantes.

La conception est de type « 2*2 » avec la particularité que nous avons ici un facteur avec trois traitements (Représentation).

Subject	JRefractory			Nutch		
	UMLNotation	Dong	SPE	UMLNotation	Dong	SPE
1						
2						
3						

Tableau 9.6 – Configuration des attributions des couples (*framework, représentation*)

Nous généralisons les besoins en équilibrage et en randomisation avant d'appliquer ces contraintes pour obtenir la configuration de l'expérience.

Généralisation de la randomisation et de l'équilibrage Pour avoir une entropie de la distribution des traitements aux différents sujets, nous généralisons ce problème avant de l'appliquer au cas de notre expérience.

Soit :

A L'ensemble des sujets

F L'ensemble des frameworks (traitements)

R L'ensemble des représentations (traitements)

N Le nombre d'attributions données à chaque sujet ($\langle \text{framework}, \text{représentation} \rangle$)

$f : A \rightarrow F * S * R$ fonction qui retourne, pour un sujet donné S , l'ensemble de ses attributions.

Une attribution est un couple ($\text{framework}, \text{représentation}$) $\in F \times R$.

Le nombre de combinaison d'attributions pour un sujet est égale à $(\#F * \#R)^N$.

L'objectif d'entropie pose plusieurs contraintes, en supposant que $\#F > 1$ et $\#R > 1$:

C1 : Chacune des attributions d'un sujet doit concerner un framework différent :

$$\forall a \in A \quad \forall (f_1, s_1, r_1), (f_2, s_2, r_2) \in f(a) : (f_1, s_1, r_1) \neq (f_2, s_2, r_2) \Rightarrow p_1 \neq p_2$$

C2 : Chacune des attributions d'un sujet doit concerner une représentation différente :

$$\forall a \in A \quad \forall (f_1, s_1, r_1), (f_2, s_2, r_2) \in f(a) : (f_1, s_1, r_1) \neq (f_2, s_2, r_2) \Rightarrow r_1 \neq r_2$$

Configuration de l'expérience Il est possible d'établir une configuration pour l'attribution de couples ($\text{framework}, \text{patron}$) aux sujets. L'ensemble des traitements F contient deux éléments : JRefractory et Nutch. L'ensemble des traitements E contient trois éléments : UMLNotation, Dong et SPE. Pour éviter une durée d'expérience trop élevée, nous avons décidé de fixer le nombre d'attributions N à deux. En conclusion, **pour que chaque représentation soit assignée au moins une fois, le nombre de sujets minimum est de trois**. Le tableau 9.6 présente la distribution des couples ($\text{framework}, \text{représentation}$) qu'il est nécessaire de respecter afin de réduire l'effet d'apprentissage des sujets (section 14.4).

Évaluation de la validité

Les menaces à la validité sont présentées en section 14.4.

9.2.3 Exécution

L'expérience se déroule comme présenté en section 9.2.3. Chaque sujet se voit attribuer une représentation (UMLNotation, Dong, SPE) de chacun des deux frameworks (JRefractory et Nutch). Pour chaque diagramme qui lui est attribué, six questions lui sont posées. Deux portant sur la tâche

« composition », deux autres sur la tâche « participation » et enfin les deux dernières sur la tâche « rôle ».

Vu la particularité de notre expérience, une étape préliminaire servant à calibrer l'eye-tracker suivant le sujet était nécessaire. En effet, sans cette étape des offsets trop grands peuvent apparaître et empêcher l'utilisation des données récoltées.

Chaque framework est introduit par une petite description du projet comme visible dans le questionnaire présenté en AnnexeB. Une fois que le sujet a lu cette description, il peut prendre connaissance de la première question. Dès que le sujet a compris la question et qu'il est prêt, il peut afficher le diagramme. Dans un premier temps, le sujet recherche les données nécessaires pour répondre à la question. Une fois qu'il a identifié les éléments, qui pour lui, répondent à la question, il passe au diagramme du support (qui est identique au diagramme de question) et peut répondre sur sa feuille. Ce qui signifie qu'il peut toujours utiliser le diagramme comme support. Les données récoltées durant cette partie de la tâche serviront aux métriques utilisées pour le rassemblement d'information (support). Tandis que les données récoltées dans la première partie de la tâche serviront pour la recherche d'informations (cf. section 9.1.8). Quand toutes les questions ont été répondues, le deuxième framework est présenté au sujet. Le fonctionnement est identique d'un framework à l'autre. La durée de l'expérience est calibrée pour ne pas excéder l'heure.

Étapes de l'expérience

L'expérimentation se déroulera dans une pièce « calme » sans aucune distraction. Le passage d'un sujet se déroulera comme suit :

1. Le sujet s'enregistre personnellement pour l'étude, par mail, ou en utilisant le lien doodle d'inscription.
2. Nous présentons un tutoriel pour introduire les trois représentations utilisées lors de l'expérimentation avec quelques diagrammes d'exemples pour familiariser le sujet avec les tâches qu'il va réaliser plus tard. Ce tutoriel est utile pour atténuer l'anxiété et pour donner à tous les sujets la même connaissance de base nécessaire pour utiliser les représentations. Ce tutoriel sert aussi à réduire l'« effet d'apprentissage ».
3. Dès la fin du tutoriel, nous donnons une présentation de l'eye tracker utilisé pour collecter les données et les instructions importantes d'utilisation du système (par exemple : éviter les mouvements de la tête durant l'exécution des tâches).
4. Nous installons le sujet sur une chaise de dentiste fixée au sol et nous effectuons un test de calibration. Nous procédons ensuite à une simulation afin de nous assurer que cette étape s'est bien déroulée.
5. Le sujet peut ensuite prendre connaissance de la mise en contexte du framework.
6. Une fois prêt, le sujet peut lire la première question et s'assurer de bien l'avoir comprise.
7. Dès que le sujet a compris la question, il appuie sur la touche ENTER du clavier, le diagramme est affiché et l'enregistrement commence.
8. Si le sujet pense avoir trouvé la réponse, celui-ci appuie sur la touche ENTER du clavier et écrit sa réponse sur papier. Le diagramme est toujours présent à titre de support afin que le sujet n'ait pas à retenir la réponse par cœur. L'enregistrement continue durant cette période afin de pouvoir analyser l'effort qui est produit par le sujet pour rassembler les éléments de la réponse une fois qu'il pense avoir trouvé celle-ci.

9. Lorsque le sujet a fini de répondre, il appuie sur ENTER et peut commencer à lire la deuxième question.
10. Le processus continue de manière analogue jusqu'à ce que toutes les questions soient répondues.
11. Une fois que toutes les questions d'un type de représentation sont terminées, il est demandé à l'utilisateur d'évaluer sa charge mentale subjective à l'aide d'un questionnaire NASA-TLX.
12. Finalement, un questionnaire post-expérimental est donné au sujet pour évaluer ses connaissances en patron de conception et en UML.

Analyse de la tâche « composition »

Après avoir collecté les données expérimentales durant la phase d'opération de l'expérience, nous sommes en mesure d'analyser et d'interpréter les résultats. Suivant la conception de l'expérience présentée dans le Chapitre 9, nous allons décomposer l'analyse des résultats en quatre parties : une pour chaque tâche (« composition », « participation » et « rôle ») et une dernière pour les mesures subjectives.

Ce chapitre se concentre donc uniquement sur la tâche d'identification des patrons de conception dans lesquels une classe donnée intervient (tâche « composition »). Toutes les analyses présentes dans ce chapitre, et dans les chapitres suivants, ont été réalisées avec le logiciel R en suivant les recommandations établies par les ouvrages de Wholin et al. [WRH+00] et de Juristo et al. [JM10].

La section 10.1 pose le contexte et l'approche adoptée pour les analyses des résultats. La deuxième partie du chapitre (section 10.2) contient la sélection des métriques significatives pour les trois axes d'analyses que nous avons choisis d'adopter : l'analyse de la performance, l'analyse des mesures physiologiques et l'analyse de la distance relative entre chemins visuels. Le détail de l'étude des mesures physiologiques est disponible en section 10.3. La section 10.4 évalue en détail l'influence des facteurs confondants (sexe et niveau d'étude) sur la distance relative entre chemins visuels. L'ensemble des conclusions de ce chapitre est ensuite synthétisé en section 10.5.

Sommaire

10.1	Structuration de l'analyse des tâches	107
10.2	Sélection des métriques significatives	108
10.2.1	Sélection des métriques de performances significatives	108
10.2.2	Sélection des métriques physiologiques significatives	108
10.2.3	Sélection des distances relatives entre chemins visuels significatives	110
10.3	Analyse des mesures physiologiques	110
10.3.1	Étude de l'influence de la représentation : IN AORI/IN AOII normalisé	110
10.3.2	Étude de l'influence de la représentation : ratio durée fixation/saccade	112
10.3.3	Étude de l'influence de la représentation : pourcentage de transition AORI vers AOII	113
10.4	Analyse des distances relatives entre chemins visuels	115
10.4.1	Données	115
10.4.2	Hypothèses	116
10.4.3	Tests d'hypothèses	116
10.5	Conclusion de l'analyse de la tâche	116

10.1 Structuration de l'analyse des tâches

L'analyse des résultats de chaque tâche est conçue suivant les trois mêmes axes : une *analyse de la performance*, une *analyse des mesures physiologiques* et une *analyse de la distance relative entre chemins visuels*. Comme nous l'avons précédemment énoncé au sein de la section 9.1.8, pour chaque tâche réalisée par le sujet, nous avons différencié la partie se rapportant à la *recherche des éléments de réponses* (nommée « *question* » dans le reste du document) et la partie se rapportant au *rassemblement des éléments de réponses* (nommée *support* dans le reste du document). Cependant, pour des raisons pratiques, cette différenciation est uniquement réalisée au sein de l'analyse des mesures physiologiques. L'étude des performances et l'étude de la distance entre parcours visuels ne couvrent donc que la partie « question¹ ».

Les deux premiers axes, l'analyse de la performance et l'analyse des mesures physiologiques, étudient l'influence qu'ont les variables indépendantes sur les variables dépendantes. Enfin, une analyse de l'impact des facteurs confondants sur la distance relative entre chemins visuels est traitée dans le dernier axe d'analyse.

L'analyse de la performance est basée sur le temps passé sur la tâche et sur l'exactitude des réponses fournies par le sujet.

Les métriques utilisées pour le deuxième axe d'étude sont : la *densité spatiale*, la *densité de transition*, le *taux d'IN AORI/IN AOII normalisé*, la *durée moyenne des fixations*, le *ratio de durée fixation/saccade*, le *pourcentage de transition AORI vers AOII* et la *nombre moyen de fixations post-AORI*.

Le dernier axe utilise la métrique *distance relative entre chemins visuels*, proposée par De Smet et Lempereur [DL11] et adaptée à DTW par nos soins. Une présentation de cette métrique a été précédemment réalisée en section 8. Les données utilisées, pour ce dernier axe, se rapportent à la moyenne calculée à partir des distances des chemins visuels d'un groupe. Dès lors, les analyses ne se concentrent pas sur les différents types de représentations, mais sur les différentes façons qu'utilisent les groupes pour parcourir un diagramme UML. Ces groupes que nous avons analysés sont des groupes formés à partir du sexe et du niveau d'étude (c'est-à-dire master ou doctorant). Dans la suite de ce document, le groupe « Master » contiendra toute personne ayant comme diplôme le plus élevé un master ou toute personne effectuant des études en vue de l'obtention du diplôme de master. Toute personne ayant comme diplôme le plus élevé un doctorat ou toute personne effectuant des études en vue de l'obtention de ce diplôme fera partie du groupe « Doctorant ».

Lors de l'exécution de l'expérience, nous avons remarqué que les sujets produisaient beaucoup de mouvements de la tête lors de la partie support des trois tâches. Ces mouvements risquent de perturber l'enregistrement des saccades, utiles pour certaines métriques. Par conséquent, seules les mesures se rapportant aux fixations seront gardées pour la partie support de l'axe d'analyse des mesures physiologiques. En conclusion seules les métriques : densité spatiale, taux d'IN AORI/IN AOII normalisé et durée moyenne des fixations ont été enregistrées pour la partie « question » ainsi que pour la partie support. Pour les autres métriques, les données collectées ne se rapportent qu'à la partie « question » de la tâche.

Afin de faciliter la lecture et la compréhension de chaque chapitre, l'examen de ceux-ci commence par une présentation des résultats des tests d'hypothèses pour les trois axes d'analyses. Les analyses

1. Dans le reste de ce document, si la partie de la tâche (question ou support) n'est pas précisée, le lecteur doit considérer qu'il s'agit de la partie question.

	Représentation	Framework
Exactitude	0.7325	1
Temps	0.5054	0.8826

Tableau 10.1 – Résumé des tests d’hypothèses des mesures de performance en fonction des variables indépendantes pour la tâche « composition »

contenues dans cette partie identifient la présence de différences entre les échantillons, mais ne déterminent pas quelles sont ces différences. Ainsi, si un axe d’analyse révèle des influences significatives, celles-ci seront étudiées en détail dans une section leur étant propre. Comme nous l’avons déjà énoncé dans la sous-section 6.3.5, l’analyse de facteurs à plus de deux traitements comporte une particularité. Si une variable indépendante à plus de deux traitements montre une influence significative, il est alors nécessaire de reproduire un test d’hypothèse par paire de traitements, afin de déterminer quelles paires possèdent une différence. Ces tests, que l’on nomme ici *tests de paires de traitements*, seront formulés dans les sections qui leurs sont propres.

Comme il est commun de le faire au sein des expériences en génie logiciel, nous avons fixé la variable α à 0.05 pour l’ensemble des tests d’hypothèses utilisés dans ce document. Néanmoins, certains résultats, dont la p-value est comprise entre 0.05 et 0.1, seront mis en avant du fait qu’ils sont presque significatifs.

10.2 Sélection des métriques significatives

10.2.1 Sélection des métriques de performances significatives

Cette sous-section contient le résultat des tests d’hypothèses réalisés sur les mesures de performance, à savoir le temps passé sur une tâche et l’exactitude des réponses. Le tableau 10.1 présente les *p-values* résultantes des tests d’hypothèses appliquées aux données de l’expérience. Ces valeurs ont été calculées grâce aux tests Kruskal-Wallis (pour la représentation) et Mann-Whitney (pour le framework). Le tableau permet d’évaluer l’influence des variables indépendantes (représentation et framework) sur les variables dépendantes. Une p-value inférieure à $\alpha = 0.05$ confirme la présence d’une influence significative.

D’après les valeurs contenues dans le tableau, il est impossible de relever une influence significative des variables indépendantes sur l’exactitude des réponses ou sur le temps nécessaire au sujet pour répondre aux questions. Cependant, il est possible d’en conclure une non-influence sur la variable d’exactitude des réponses. En effet, seule une valeur, calculée à partir de l’unité à laquelle on soustrait la p-value (1-pvalue), inférieure à α nous autorise à tirer cette conclusion. Or, la p-value du framework (p-value= 1) montre qu’il n’influence pas la variable dépendante d’exactitude. Il est donc possible de conclure que :

Conclusion 1. *Pour la tâche « composition », l’exactitude des réponses n’est pas influencée par le framework utilisé.*

10.2.2 Sélection des métriques physiologiques significatives

Le résultat des tests d’hypothèses réalisés sur les mesures physiologiques est présenté dans cette sous-section. Seules les p-values du ratio de fixation/saccade ont été calculées avec le test ANOVA. Le reste des p-values proviennent des tests Kruskal-Wallis (pour la représentation) et Mann-Whitney

Mesure	Question/Support	Représentation	Framework
Densité spatiale	Question	0.188	0.9144
	Support	0.7167	0.08734
Densité de transition	Question	0.7636	0.6004
	Support	-	-
IN AORI/IN AOII normalisé	Question	0.02731	0.6005
	Support	0.1723	0.6458
Durée moyenne des fixations	Question	0.229	0.2888
	Support	0.9043	0.9839
Ratio durée fixation/saccade	Question	0.0105	0.8546
	Support	-	-
Pourcentage de transition AORI vers AOII	Question	0.003601	0.7213
	Support	-	-
Nombre moyen de fixations post-AORI	Question	0.9722	0.9679
	Support	-	-

Tableau 10.2 – Résumé des p-values des mesures physiologiques en fonction des variables indépendantes pour la tâche « composition »

(pour le framework). Le tableau 10.2 contient le résumé des tests d'hypothèses des mesures physiologiques pour la tâche « composition ». Le détail des résultats de ces analyses se trouve dans la section 10.3.

Pour la représentation, plusieurs p-values ont une valeur inférieure à $\alpha = 0.05$: IN AORI/IN AOII normalisé (« question »), ratio durée fixation/saccade et pourcentage de transition AORI vers AOII. Nous pouvons également relever le fait que la mesure du nombre moyen de fixations post-AORI n'est pas influencée par la représentation.

En ce qui concerne le framework, le tableau ne présente aucune valeur inférieure à $\alpha = 0.05$. Néanmoins, la densité spatiale de la partie support possède une valeur comprise entre 0.05 et 0.1 que nous décidons de mettre en avant du fait qu'elle est presque significative. Cette métrique est donc légèrement influencée par le framework utilisé. Deux p-values expriment une non-influence du framework sur les variables : durée moyenne des fixations (support) et nombre moyen de fixations post-AORI.

Plusieurs conclusions peuvent donc être tirées :

Conclusion 2. *Pour la partie « question » de la tâche « composition », les variables : IN AORI/IN AOII normalisé, ratio durée fixation/saccade et pourcentage de transition AORI vers AOII sont influencées par la représentation utilisée.*

Conclusion 3. *Pour la tâche « composition », la variable du nombre moyen de fixations post-AORI n'est pas influencée par la représentation utilisée et par le framework utilisé.*

Conclusion 4. *Pour la partie « support » de la tâche « composition », la densité de transition est légèrement influencée par le framework utilisé.*

Conclusion 5. *Pour la partie « support » de la tâche « composition », la variable durée moyenne des fixations n'est pas influencée par le framework utilisé.*

	Sexe	Niveau d'étude
DRCV	0.113	0.0006982

Tableau 10.3 – Résumé des tests d'hypothèses sur la mesure de la distance relative entre les chemins visuels en fonction des facteurs confondants pour la tâche « composition »

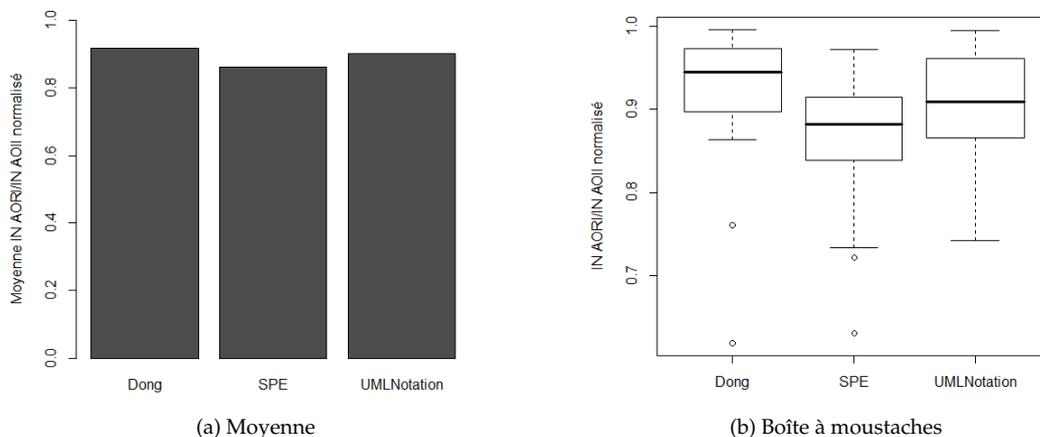


FIGURE 10.1 – Étude du taux IN AORI/IN AOII normalisé en fonction des représentations

10.2.3 Sélection des distances relatives entre chemins visuels significatives

Le tableau 10.3 inclut les p-values des tests d'hypothèses de nos deux groupes, dont le premier est un rassemblement effectué par rapport au sexe et le second par rapport au niveau d'étude (comme expliqué en section 10.1), sur la métrique de distance relative entre chemins visuels. Les tests Mann-Whitney et ANOVA ont été utilisés respectivement pour le groupe du sexe et pour le groupe du niveau d'étude. Le détail de ces analyses pour la tâche « composition » se trouve au sein de la section 10.4. Seul le niveau d'étude possède une influence significative sur la distance relative entre les chemins visuels avec un $\alpha = 0.05$. La conclusion suivante peut donc être tirée :

Conclusion 6. *Pour la tâche « composition », la distance relative entre chemins visuels est influencée par le niveau d'étude des sujets.*

10.3 Analyse des mesures physiologiques

Cette section présente l'analyse détaillée de la conclusion 2, c'est-à-dire l'étude de l'influence de la représentation sur les mesures : IN AORI/IN AOII normalisé, ratio durée fixation/saccade et le pourcentage de transition AORI vers AOII. L'analyse du framework n'étant pas l'objet d'étude de ce présent document, l'influence du framework relevée en conclusion 4 ne sera pas approfondie.

10.3.1 Étude de l'influence de la représentation : IN AORI/IN AOII normalisé

Données

Les différentes données sont détaillées dans le tableau A.3. La figure 10.1a présente un graphique des moyennes du taux IN AORI/IN AOII normalisé par représentation. La figure 10.1b illustre la répartition des données suivant la représentation pour cette métrique.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
IN AORI/IN AOII normalisé	0.005895	0.1451	0.2174

Tableau 10.4 – P-value du taux IN AORI/IN AOII normalisé pour la partie « question » de la tâche « composition » en fonction des représentations

Hypothèses

H_{0a}	Le taux IN AORI/IN AOII normalisé est, en moyenne, le même pour les sujets utilisant Dong et SPE.
H_{0b}	Le taux IN AORI/IN AOII normalisé est, en moyenne, le même pour les sujets utilisant UMLNotation et SPE.
H_{0c}	Le taux IN AORI/IN AOII normalisé est, en moyenne, le même pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que ceux utilisant Dong.

Tests d'hypothèses

Le tableau 10.4 expose le test d'hypothèse Mann-Whitney appliqué aux différentes paires de traitements, pour la partie « question » de la tâche « composition ». D'après la figure 10.1, nous pouvons remarquer que les sujets ont un taux IN AORI/IN AOII normalisé plus élevé avec la représentation Dong. D'après les résultats, un sujet se concentre plus sur des points pertinents lorsqu'il utilise Dong et qu'il recherche l'ensemble des patrons de conception dans lesquels intervient une classe. Cette observation respecte ce que nous pouvons attendre de la représentation Dong. En effet, celle-ci rassemble les informations des patrons de conception d'une classe au sein d'un seul stéréotype ; le sujet a donc accès à toutes ces informations dès qu'il a trouvé la classe qui l'intéresse.

Par ailleurs, avec SPE, les sujets ne semblent pas se fixer sur des points pertinents. Ce mauvais résultat trouve son explication dans une observation faite au cours de l'exécution de l'expérience. SPE présente des couleurs dans le but de faciliter l'identification des patrons de conception utilisés par les classes. Malheureusement, nous avons remarqué que de nombreux sujets avaient du mal à différencier les couleurs utilisées. Ainsi, les sujets ont dû faire face à ce problème en regardant des parties du diagramme non pertinentes pour distinguer les couleurs utilisées.

D'après le tableau, il existe une différence significative entre Dong et SPE. Seule l'hypothèse H_{0a} peut donc être rejetée. Par conséquent, il est possible d'affirmer (avec une probabilité d'erreur inférieure à 5%) :

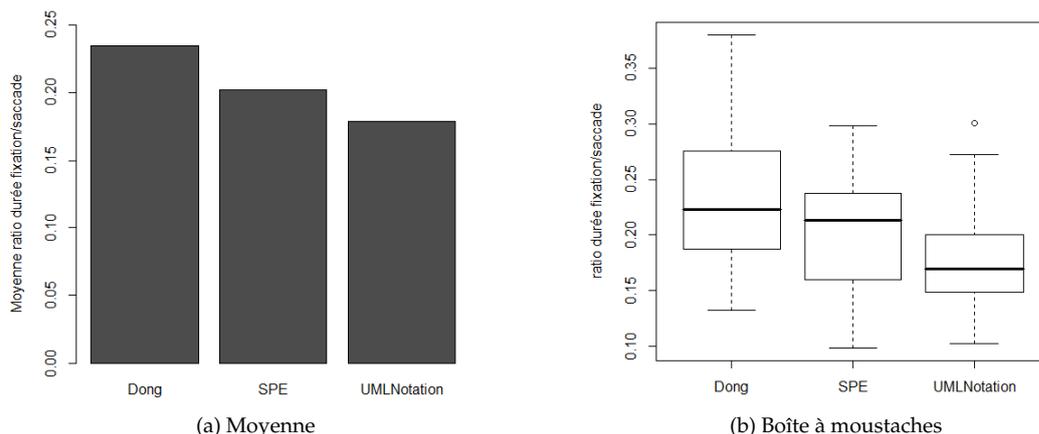


FIGURE 10.2 – Étude du ratio durée fixation/saccade en fonction de la représentation

Conclusion 7. Pour la partie « question » de la tâche « composition », les sujets utilisant Dong ont, en moyenne, un taux $IN\ AORI/IN\ AOII$ normalisé plus élevé que les sujets utilisant SPE.

10.3.2 Étude de l'influence de la représentation : ratio durée fixation/saccade

Données

Les différentes données sont détaillées dans le tableau A.3. La figure 10.2a présente un graphique des moyennes du ratio durée fixation/saccade par représentation. La figure 10.2b illustre la répartition des données suivant la représentation pour cette métrique.

Hypothèses

H_{0a}	Le ratio durée fixation/saccade est, en moyenne, le même pour les sujets utilisant Dong et SPE.
H_{0b}	Le ratio durée fixation/saccade est, en moyenne, le même pour les sujets utilisant UMLNotation et SPE.
H_{0c}	Le ratio durée fixation/saccade est, en moyenne, le même pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant Dong.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Ratio durée fixation/saccade	0.123	0.174	0.00234

Tableau 10.5 – P-value du ratio durée fixation/saccade pour la partie « question » de la tâche « composition » en fonction de la représentation utilisée

Tests d'hypothèses

Le tableau 10.5 expose le test d'hypothèse ANOVA appliqué aux trois paires. Pour la comparaison de Dong et UMLNotation, nous remarquons la présence d'un résultat significatif : p-value= 0.00234. Cette valeur inférieure à $\alpha = 0.05$ permet de rejeter l'hypothèse nulle H_{0c} .

D'après la figure 10.2a, les sujets ont un ratio de durée de fixation/saccade significativement plus élevé lorsqu'ils utilisent la représentation Dong pour rechercher l'ensemble des patrons de conception dans lesquels intervient une classe. Avec cette représentation, le temps nécessaire au traitement de l'information est équivalent à 23% du temps utilisé pour la recherche. Ce ratio plus élevé se justifie par le fait que la représentation Dong nécessite plus de traitement que la représentation UMLNotation. Lorsqu'il utilise Dong, le sujet doit rechercher la classe déterminée pour ensuite analyser le stéréotype et récupérer les informations qui l'intéressent. Par contre, avec SPE, les sujets détiennent un ratio moins élevé qu'avec Dong. Malheureusement, d'après la p-value de comparaison de ces deux représentations, cette différence n'est pas significative.

Ainsi, nous pouvons accepter l'hypothèse H_{1c-1} et tirer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 8. *Pour la partie « question » de la tâche « composition », les sujets utilisant Dong ont, en moyenne, un ratio durée de fixation/saccade significativement plus élevé que les sujets utilisant UMLNotation.*

10.3.3 Étude de l'influence de la représentation : pourcentage de transition AORI vers AOII

Données

Le tableau A.3 présente en détail les données utilisées pour cette étude. La figure 10.3a reproduit un graphique des moyennes du pourcentage de transition AORI vers AOII en fonction de la représentation. La figure 10.3b illustre la répartition des données suivant la représentation pour cette métrique.

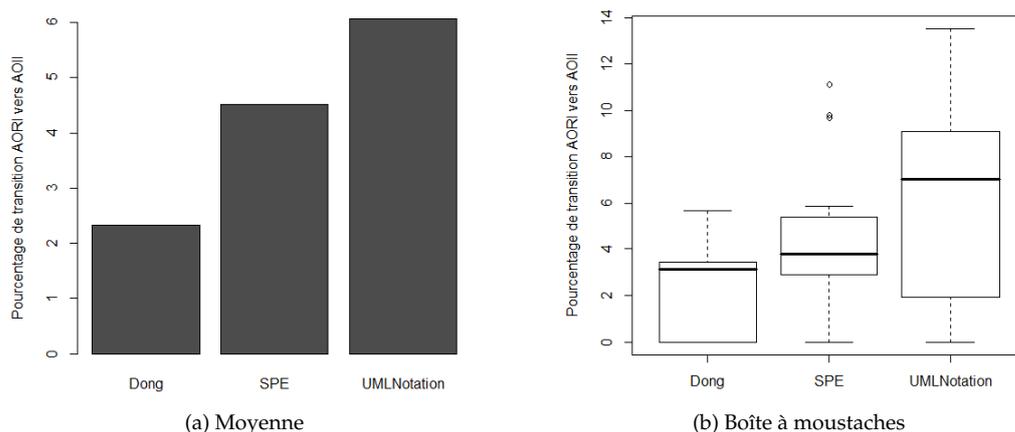


FIGURE 10.3 – Étude du pourcentage de transition AORI vers AOII en fonction de la représentation

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Transition AORI vers AOII	0.02328	0.1881	0.00165

Tableau 10.6 – P-value du pourcentage de transition AORI vers AOII pour la partie « question » de la tâche « composition » en fonction de la représentation utilisée

Hypothèses

H_{0a}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même pour les sujets utilisant Dong et SPE.
H_{0b}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même pour les sujets utilisant UMLNotation et SPE.
H_{0c}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant Dong.

Tests d'hypothèses

Le tableau 10.6 présente le résultat de l'application du test Mann-Whitney sur les paires des trois traitements. Les figures 10.3a et 10.3b laissent présager une influence de la représentation utilisée sur le pourcentage de transition de zones d'intérêts vers des zones de non-intérêts. En étant inférieures à $\alpha = 0.05$, deux p-values confirment cette observation pour la comparaison de Dong-SPE (0.02328)

et la comparaison Dong-UMLNotation (0.00165).

D'après les moyennes, le pourcentage de transition est significativement plus élevé pour les participants utilisant SPE que pour les participants utilisant Dong. À nouveau, ce phénomène est causé par le problème de différenciation des couleurs. La distinction des couleurs a nécessité, chez les sujets utilisant SPE, un va-et-vient de fixations. Pour finir, ce fait a engendré une augmentation du pourcentage de transitions.

Par ailleurs, la comparaison Dong-UMLNotation nous autorise à relever un pourcentage de transition significativement plus élevé chez les sujets utilisant UMLNotation que chez les sujets utilisant Dong. Cette dernière observation ne paraît pas, à priori, anormal. Effectivement, au sein d'UMLNotation, les informations propres aux patrons de conception sont mélangées avec les autres éléments du diagramme UML. Remarquons notamment que les lignes pointillées rejoignant les ellipses d'UMLNotation peuvent être facilement confondues avec les relations d'implémentations propres au diagramme de classes. Ceci peut donc pousser l'utilisateur à se focaliser sur des fragments de diagramme sans intérêt et, par conséquent, augmenter le pourcentage de transition AORI vers AOII.

Les p-value nous permettent de rejeter H_{0a} et H_{0c} et d'établir les deux conclusions suivantes (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 9. *Pour la partie « question » de la tâche « composition », les sujets utilisant SPE ont un pourcentage de transition AORI vers AOII significativement plus élevé que les sujets utilisant Dong.*

Conclusion 10. *Pour la partie « question » de la tâche « composition », les sujets utilisant UMLNotation ont un pourcentage de transition AORI vers AOII significativement plus élevé que les sujets utilisant Dong.*

10.4 Analyse des distances relatives entre chemins visuels

Cette section a pour objectif de présenter en détail les résultats énoncés dans la sous-section 10.2.3. Comme nous l'avons remarqué dans cette sous-section, le sexe des participants n'influence en rien la métrique étudiée. Par conséquent, le niveau d'étude des participants est le seul facteur confondant pris en compte dans cette section.

10.4.1 Données

Les données utilisées pour cette analyse sont reprises en détails au sein du tableau A.6. Les moyennes des distances relatives entre chemins visuels en fonction du niveau d'étude sont illustrées par le graphique de la figure 10.4a. La figure 10.4b présente également la répartition des distances en fonction du niveau d'étude.

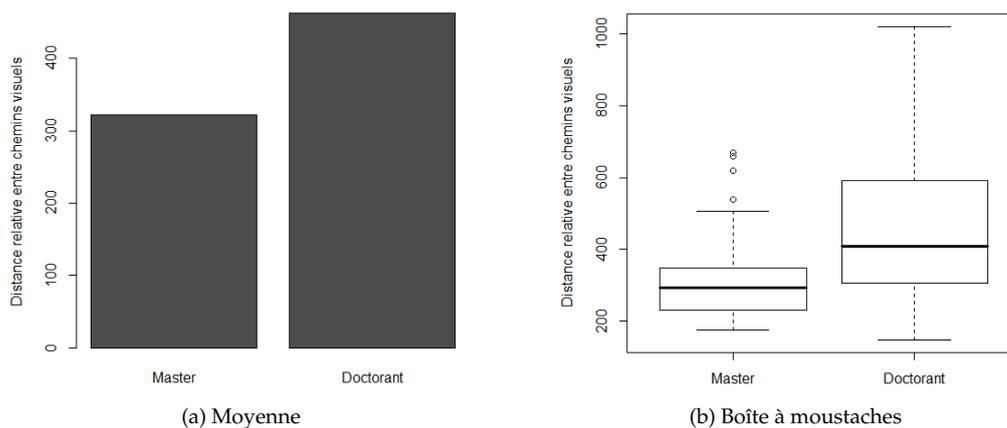


FIGURE 10.4 – Étude de la distance relative entre chemins visuels en fonction du niveau d'étude des participants pour la tâche « composition »

10.4.2 Hypothèses

H_0 La distance relative entre chemins visuels est, en moyenne, la même quel que soit le niveau d'étude des participants.

H_1 La distance relative entre chemins visuels, est, en moyenne, plus faible pour le niveau d'étude X .

$X \in \{\text{Master, Doctorant}\}$

10.4.3 Tests d'hypothèses

Pour tester l'hypothèse nulle, le test non-paramétrique Mann-Whitney a été utilisé. La figure 10.4a révèle concrètement que les doctorants présentent une moyenne de distance relative entre chemins visuels plus élevée. Le résultat du test d'hypothèse ($= 0.0006982$) nous permet même de dire que cette influence est significative. La boîte à moustaches contenue dans la figure 10.4b confirme cette observation. En effet, les distances relatives entre chemins visuels chez les doctorants sont fortement dispersées sur l'ensemble du graphique. La tendance centrale des masters est, quant à elle, très compacte. Ceci montre qu'ils semblent se comporter de façon plus homogène que les doctorants. La p-value résultante nous autorise à rejeter H_0 au profit de H_1 et à conclure (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 11. *Pour la partie « question » de la tâche « composition », les masters ont, en moyenne, une distance relative entre chemins visuels significativement plus faible que les doctorants.*

10.5 Conclusion de l'analyse de la tâche

L'analyse de la tâche de composition de patron, consistant à nommer les patrons dans lesquels une classe déterminée participe, permet de tirer plusieurs conclusions. La non-influence des frameworks sur l'exactitude des réponses est la seule conclusion qui ressort de l'axe de la performance (conclusion 1).

L'analyse des métriques propres aux données fournies par l'oculomètre nous autorise à tirer des conclusions relatives à l'influence ou à la non-influence de certaines variables indépendantes. Concernant la non-influence, nous pouvons énoncer que :

- Le nombre moyen de fixations post-AORI n'est pas influencée par la représentation et le framework utilisé (conclusion 3).
- Lorsqu'un sujet réalise une tâche de composition de patron et qu'il rassemble les éléments de réponse, la durée moyenne de ses fixations n'est pas influencée par le framework utilisé (conclusion 5).

En cas de recherche des éléments de réponse par le sujet, une influence de la représentation utilisée est notable pour trois des sept métriques étudiées. D'après les résultats, un sujet utilisant Dong et recherchant l'ensemble des patrons de conception dans lesquels une classe intervient, se concentrera plus sur des points pertinents du diagramme que les sujets utilisant SPE (conclusion 7). Dans ce même cas, la proportion durée du traitement/durée de la recherche de l'information sera significativement plus élevée pour les sujets utilisant Dong que pour les sujets utilisant UMLNotation (conclusion 8). Pour Dong, nous pouvons également dire que le temps de traitement nécessaire au sujet sera en moyenne équivalent à 23 % du temps de recherche. D'après ces éléments, il semble qu'avec Dong, les sujets repèrent plus facilement les stéréotypes contenus dans les classes, mais que l'extraction de l'information contenue dans ces stéréotypes nécessite plus de temps de traitement.

Les utilisateurs d'UMLNotation ou de SPE ont un taux de transition des zones d'intérêts vers les zones de non-intérêts significativement plus élevé que les sujets utilisant Dong (conclusions 9 et 10). Il s'avère donc que les utilisateurs des représentations UMLNotation et SPE ont une mauvaise interprétation des zones d'intérêt lorsqu'ils recherchent les patrons de conception dans lesquels une classe donnée intervient. Cette observation soulève un inconvénient d'UMLNotation. En effet, cette dernière mélange les informations des patrons de conception avec les autres éléments du diagramme de classes. Ainsi, ce désavantage peut entraîner le sujet à se focaliser sur des fragments de diagramme non pertinents. Par ailleurs, le résultat élevé avec SPE pour la métrique de transition AORI vers AOII est causée par une difficulté de différenciation des couleurs par les sujets.

Enfin, la conclusion 11 manifeste un impact du niveau d'étude sur la distance relative entre parcours visuels. Dans une expérience différente de la nôtre, cette constatation avait déjà été faite par De Smet et Lempereur [DL11]. Il semble donc que les masters se comportent de façon plus homogène que les doctorants.

Analyse de la tâche « participation »

La deuxième partie de l'analyse des résultats concerne la tâche de participation des classes. Au cours de cette tâche, le sujet doit identifier toutes les classes participant à un patron de conception particulier. La structuration des analyses utilisée au sein de ce chapitre est la même que celle décrite en section 10.1. Pour rappel, nous analysons les résultats sur base de trois axes : analyse de la performance, analyse des mesures physiologiques et analyse de la distance relative entre chemins visuels.

La section 11.1 contient la sélection des métriques significatives de ces trois axes. L'étude détaillée du premier axe d'analyse est présent en section 11.2. La section 11.3 approfondit l'analyse de l'influence des représentations sur les mesures oculométriques. L'impact du sexe et du niveau d'études des sujets sur la distance relative entre chemins visuels est présenté en section 11.4. Pour finir, la section 11.5 conclut l'analyse des résultats de la tâche de participation des classes.

Sommaire

11.1	Sélection des métriques significatives	121
11.1.1	Sélection des métriques de performance significatives	121
11.1.2	Sélection des métriques physiologiques significatives	121
11.1.3	Sélection des distances relatives entre chemins visuels significatives	123
11.2	Analyse des performances	123
11.2.1	Étude de l'influence de la représentation : temps passé sur une tâche	123
11.2.2	Étude de l'influence de la représentation : validité des réponses	125
11.3	Analyse des mesures physiologiques	126
11.3.1	Étude de l'influence de la représentation : densité spatiale	126
11.3.2	Étude de l'influence de la représentation : densité de transition	128
11.3.3	Étude de l'influence de la représentation : IN AORI/IN AOII normalisé - partie question	129
11.3.4	Étude de l'influence de la représentation : IN AORI/IN AOII normalisé - partie support	132
11.3.5	Étude de l'influence de la représentation : ratio durée fixation/saccade	133
11.3.6	Étude de l'influence de la représentation : pourcentage de transition AORI vers AOII	135
11.3.7	Étude de l'influence de la représentation : nombre moyen de fixations post- AORI	136
11.4	Analyse des distances relatives entre chemins visuels	137

11.4.1	Données	138
11.4.2	Hypothèses	138
11.4.3	Tests d'hypothèses	138
11.5	Conclusion de l'analyse de la tâche	139

	Framework	Représentation
Exactitude	0.4344	0.08096
Temps	0.0692	0.07949

Tableau 11.1 – Résumé des tests d’hypothèses des mesures de performance en fonction des variables indépendantes pour la tâche « participation »

11.1 Sélection des métriques significatives

11.1.1 Sélection des métriques de performance significatives

Cette sous-section présente le résultat des tests d’hypothèses réalisés sur les mesures de performance, à savoir le temps passé sur une tâche et l’exactitude des réponses. Les p-values résultantes des tests d’hypothèses appliquées aux données de la tâche « participation » sont présentées dans le tableau 11.1. Seule la distribution se rapportant au temps passé sur une tâche et au framework possède un caractère normal, particularité qui nous permet d’utiliser le test ANOVA pour calculer la p-value. Le reste des valeurs du tableau a été déterminé à partir des tests Mann-Whitney (pour le framework) et Kruskal-Wallis (pour la représentation). Une valeur inférieure à $\alpha = 0.1$ permet de confirmer une influence presque significative des variables indépendantes sur les variables dépendantes.

Nous remarquons rapidement, via les valeurs 0.08096 et 0.07949, que les deux métriques sont légèrement influencées par la représentation. Une étude détaillée de cet impact sera réalisée dans la section 11.2.

Pour le framework, le tableau présente une p-value presque significative pour le temps passé sur la tâche « participation ». L’influence du framework sur les variables dépendantes n’étant pas le sujet principal de ce document, celle-ci ne sera pas approfondie en section 11.2.

Le résumé des p-values nous autorise à énoncer les conclusions suivantes :

Conclusion 12. *Pour la tâche « participation », le temps nécessaire pour rechercher les éléments de réponse est légèrement influencé par le framework utilisé.*

Conclusion 13. *Pour la tâche « participation », l’exactitude des réponses et le temps nécessaire à la réalisation de la tâche sont légèrement influencés par la représentation utilisée.*

11.1.2 Sélection des métriques physiologiques significatives

Le résultat des tests d’hypothèses, effectué sur les mesures oculométriques, est présenté dans cette sous-section. Les mesures utilisées sont : la densité spatiale, la densité de transition, le taux d’IN AORI/IN AOII normalisé, la durée moyenne des fixations, le ratio de durée fixation/saccade, le pourcentage de transition AORI vers AOII et le nombre moyen de fixations post-AORI.

Le tableau 11.2 expose les p-values d’influence du framework et de la représentation utilisés. Ces valeurs proviennent de trois tests d’hypothèses différents. En raison du caractère normal de la distribution, le test ANOVA a été utilisé sur la métrique de durée moyenne des fixations uniquement pour la partie « question » de la tâche. Les p-values restantes proviennent des tests Mann-Whitney (pour le framework) et Kruskal-Wallis (pour la représentation).

Mesure	Question/Support	Framework	Représentation
Densité spatiale	Question	0.2042	0.0002757
	Support	0.1967	0.2422
Densité de transition	Question	0.4114	0.001082
	Support	-	-
IN AORI/IN AOII normalisé	Question	0.05911	0.0001364
	Support	0.495	0.0002695
Durée moyenne des fixations	Question	0.486	0.528
	Support	0.05776	0.4146
Ratio durée fixation/saccade	Question	0.4685	0.00139
	Support	-	-
Pourcentage de transition AORI vers AOII	Question	0.661	0.05684
	Support	-	-
Nombre moyen de fixations post-AORI	Question	0.1301	0.0003089
	Support	-	-

Tableau 11.2 – Résumé des p-values des mesures physiologiques en fonction des variables indépendantes pour la tâche « participation »

La partie du tableau se rapportant au framework illustre deux influences. L'une d'entre elles est légèrement significative ($p\text{-value} < \alpha = 0.1$) pour le taux IN AORI/IN AOII normalisé, lors de la partie question, et l'autre est légèrement significative pour la durée moyenne des fixations, lors de la partie support. Aucune autre valeur nous autorise à énoncer une conclusion concernant cette variable indépendante.

Par ailleurs, les p-values, qui concernent la représentation, indiquent une influence sur plusieurs métriques. Pour la partie « question » de la tâche, la densité spatiale, la densité de transition, le taux IN AORI/IN AOII normalisé, le ratio durée fixation/saccade et le nombre moyen de fixations post-AORI sont impactées significativement avec un $\alpha = 0.05$. Toujours pour la partie question, cette seconde variable indépendante possède une légère influence (pour un $\alpha = 0.1$) sur le pourcentage de transition AORI vers AOII. La partie support révèle, quant à elle, une valeur significative (avec un $\alpha = 0.05$) de 0.0002695 pour le taux IN AORI/IN AOII normalisé.

L'ensemble de ces observations admet plusieurs conclusions :

Conclusion 14. *Pour la partie « question » de la tâche « participation », les variables : densité spatiale, densité de transition, IN AORI/IN AOII normalisé, ratio durée fixation/saccade et nombre moyen de fixations post-AORI sont influencées par la représentation utilisée.*

Conclusion 15. *Pour la partie « question » de la tâche « participation », la variable pourcentage de transition AORI vers AOII est légèrement influencée par la représentation utilisée.*

Conclusion 16. *Pour la partie « support » de la tâche « participation », la variable IN AORI/IN AOII normalisé est influencée par la représentation utilisée.*

Conclusion 17. *Pour la partie « question » de la tâche « participation », la variable IN AORI/IN AOII normalisé est légèrement influencée par le framework utilisé.*

Conclusion 18. *Pour la partie « support » de la tâche « participation », la durée moyenne des fixations est légèrement influencée par le framework utilisé.*

	Sexe	Niveau d'étude
DRCV	0.311	0.01284

Tableau 11.3 – Résumé des tests d'hypothèses sur la mesure de la distance relative entre les chemins visuels en fonction des facteurs confondants pour la tâche « participation »

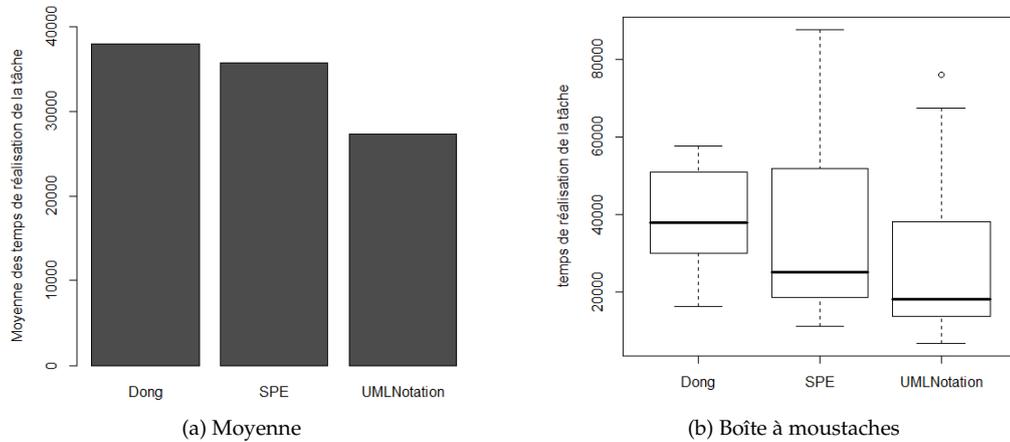


FIGURE 11.1 – Étude du passé sur la tâche « participation » en fonction de la représentation

11.1.3 Sélection des distances relatives entre chemins visuels significatives

Le tableau 11.3 contient les résultats des tests d'hypothèses réalisés sur les moyennes des distances relatives entre parcours visuels suivant le sexe et le niveau d'études des participants. La p-value du sexe des sujets résulte du test ANOVA (car les données suivent une distribution normale), tandis que celle du deuxième facteur confondant résulte du test Mann-Whitney. Seul le niveau d'étude (avec une valeur de 0.01284) possède une influence significative sur la métrique étudiée, si l'on considère un $\alpha = 0.05$. L'analyse de cette influence est approfondie en section 11.4.

Par conséquent, il est possible de conclure que :

Conclusion 19. *Pour la tâche « participation », la distance relative entre chemins visuels est influencée par le niveau d'étude des sujets.*

11.2 Analyse des performances

Cette section présente l'analyse détaillée de la conclusion 13. L'analyse du framework n'étant pas l'objet d'étude principal de ce présent document, l'influence de ce dernier, relevée en conclusion 12, ne sera pas approfondie.

11.2.1 Étude de l'influence de la représentation : temps passé sur une tâche

Données

Les données manipulées dans cette sous-section sont reprises dans le tableau A.2. Les graphiques 11.1a et 11.1b illustrent, respectivement, les moyennes et la distribution du temps nécessaire à la recherche des éléments de la tâche de participation des classes.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Temps passé sur la tâche	0.5007	0.2645	0.02483

Tableau 11.4 – P-value du temps passé sur la tâche « participation » en fonction de la représentation utilisée

Hypothèses

H_{0a}	Les sujets utilisent, en moyenne, le même temps avec Dong et SPE.
H_{0b}	Les sujets utilisent, en moyenne, le même temps avec UMLNotation et SPE.
H_{0c}	Les sujets utilisent, en moyenne, le même temps avec Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont besoin, en moyenne, d'un temps significativement plus faible que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont besoin, en moyenne, d'un temps significativement plus faible que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont besoin, en moyenne, d'un temps significativement plus faible que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont besoin, en moyenne, d'un temps significativement plus faible que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont besoin, en moyenne, d'un temps significativement plus faible que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont besoin, en moyenne, d'un temps significativement plus faible que ceux utilisant Dong.

Tests d'hypothèses

D'après le graphique des moyennes (figure 11.1a), avec UMLNotation, les sujets ont besoin de moins de temps qu'avec les deux autres représentations. Le deuxième graphique, la boîte à moustaches, nous révèle des informations supplémentaires. Sur ce graphique, SPE possède la tendance centrale la plus étendue des trois représentations, c'est-à-dire qu'avec SPE, la recherche des réponses requiert aux sujets un temps fortement variable. Seulement, d'après les p-values de comparaison des traitements (tableau 11.4), aucune différence significative n'est trouvée avec SPE.

Par contre, la comparaison de Dong et UMLNotation affirme une différence significative pour un $\alpha = 0.05$. Dès lors, les sujets utilisant UMLNotation ont besoin de moins de temps pour rechercher les éléments de réponses de la tâche « participation », que ceux utilisant Dong. Nous pouvons par contre relever le fait qu'UMLNotation présente une boîte à moustaches plus étendue que celle de Dong. D'après cette observation, le temps nécessaire à la réalisation de la tâche est donc plus variable chez les sujets utilisant UMLNotation.

La p-value significative nous autorise à rejeter l'hypothèse H_{0c} et à affirmer (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 20. *Pour la partie « question » de la tâche « participation », les sujets utilisant UML-Notation ont besoin d'un temps significativement plus faible que ceux utilisant Dong.*

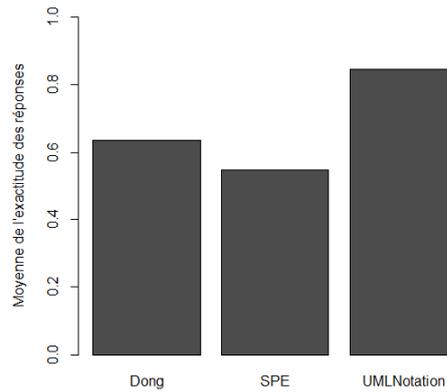


FIGURE 11.2 – Moyenne de l'exactitude des réponses de la tâche « participation » en fonction de la représentation

11.2.2 Étude de l'influence de la représentation : validité des réponses

Données

La figure 11.2 présente, au lecteur, les moyennes des taux de bonnes réponses en fonction de la représentation. Les données utilisées sont issues du tableau A.2.

Hypothèses

H_{0a}	Le taux de réponses exactes est, en moyenne, le même, pour les sujets utilisant Dong et SPE.
H_{0b}	Le taux de réponses exactes est, en moyenne, le même, pour les sujets utilisant UMLNotation et SPE.
H_{0c}	Le taux de réponses exactes est, en moyenne, le même, pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant Dong.

Tests d'hypothèses

Suivant les moyennes illustrées par le graphique 11.2, les sujets ont un meilleur taux de bonnes réponses lorsqu'ils utilisent UMLNotation. Nous remarquons que la p-value de comparaison UMLNotation-SPE du tableau 11.1 (= 0.02978) est significative pour un $\alpha = 0.05$. Nous pouvons donc rejeter H_{0b} et valider l'hypothèse H_{1b-1} (avec une probabilité d'erreur inférieure à 5%) :

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Validité des réponses	0.5838	0.02978	0.1008

Tableau 11.5 – P-value du caractère exact des réponses de la tâche « participation » en fonction de la représentation utilisée

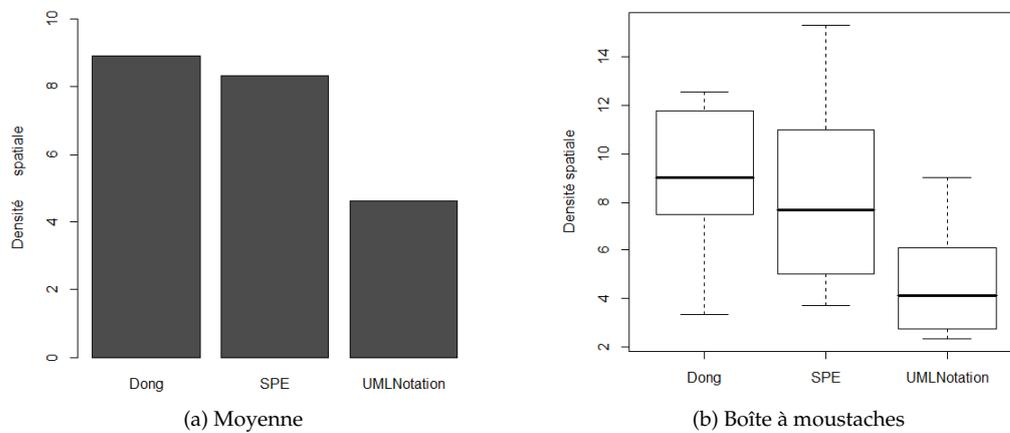


FIGURE 11.3 – Étude de la densité spatiale en fonction de la représentation

Conclusion 21. *Pour la partie « question » de la tâche « participation », les sujets utilisant UML-Notation possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant SPE.*

11.3 Analyse des mesures physiologiques

Cette section contient l'étude approfondie des conclusions 14, 15 et 16. Par ailleurs, les conclusions propres à l'influence du framework (conclusions 17 et 18) ne seront pas examinées en détail.

11.3.1 Étude de l'influence de la représentation : densité spatiale

Données

Les graphiques 11.3a et 11.3b illustrent les moyennes de la densité spatiale et les boîtes à moustaches de la densité spatiale suivant la représentation utilisée. Ces deux figures présentent les données contenues dans le tableau A.3.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Densité spatiale	0.4934	0.006227	0.0001992

Tableau 11.6 – P-value de la densité spatiale pour la tâche « participation » en fonction de la représentation utilisée

Hypothèses

H_{0a}	La densité spatiale est, en moyenne, la même pour les sujets utilisant Dong et SPE.
H_{0b}	La densité spatiale est, en moyenne, la même pour les sujets utilisant UML-Notation et SPE.
H_{0c}	La densité spatiale est, en moyenne, la même pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, une densité spatiale plus faible que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, une densité spatiale plus faible que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, une densité spatiale plus faible que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, une densité spatiale plus faible que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, une densité spatiale plus faible que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, une densité spatiale plus faible que ceux utilisant Dong.

Tests d'hypothèses

Comme la figure 11.3a peut le montrer, les sujets possèdent une densité spatiale plus faible avec UMLNotation. Ainsi, les sujets ayant utilisés UMLNotation ont focalisé leur regard dans une plus petite zone, et ont donc procédé à une recherche concentrée et efficiente. En effet, la structure de la représentation offre cet avantage pour la tâche « participation ». Dès que l'ellipse d'un patron de conception a été trouvée, le sujet peut aisément porter son regard sur les classes qui y sont reliées. Par ailleurs, avec Dong et SPE, les sujets doivent parcourir l'ensemble du diagramme pour sélectionner toutes les classes du patron de conception, ce qui engendre une augmentation de la densité spatiale.

Le tableau 11.2 contient les p-values, de comparaison des paires de traitements, résultantes du test Mann-Whitney. Plus formellement, les p-values de comparaison de UMLNotation-SPE (= 0.006227) et Dong-UMLNotation (= 0.0001992) démontrent une influence significative d'UMLNotation.

Ces éléments nous autorisent à rejeter les hypothèses H_{0b} et H_{0c} et à affirmer H_{1b-1} et H_{1c-2} (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 22. *Pour la partie « question » de la tâche « participation », les sujets utilisant UML-Notation ont, en moyenne, une densité spatiale significativement plus faible que ceux utilisant SPE ou Dong.*

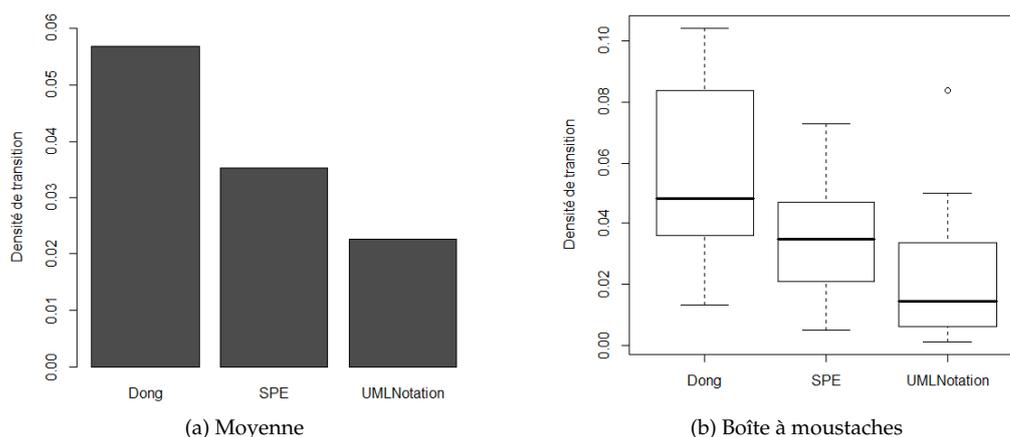


FIGURE 11.4 – Étude de la densité de transition en fonction de la représentation

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Densité de transition	0.05183	0.05358	0.0005729

Tableau 11.7 – P-value de la densité de transition pour la tâche « participation » en fonction de la représentation utilisée

11.3.2 Étude de l'influence de la représentation : densité de transition

Données

Les différentes données qui concernent la densité de transition pour la tâche « participation » sont présentes dans le tableau A.3. Les figures 11.4a et 11.4b illustrent, respectivement, les moyennes et la distribution des données pour la métrique étudiée selon la représentation.

Hypothèses

H_{0a}	La densité de transition est, en moyenne, la même avec Dong et SPE.
H_{0b}	La densité de transition est, en moyenne, la même avec UMLNotation et SPE.
H_{0c}	La densité de transition est, en moyenne, la même avec Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, une densité de transition plus faible que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, une densité de transition plus faible que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, une densité de transition plus faible que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, une densité de transition plus faible que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, une densité de transition plus faible que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, une densité de transition plus faible que ceux utilisant Dong.

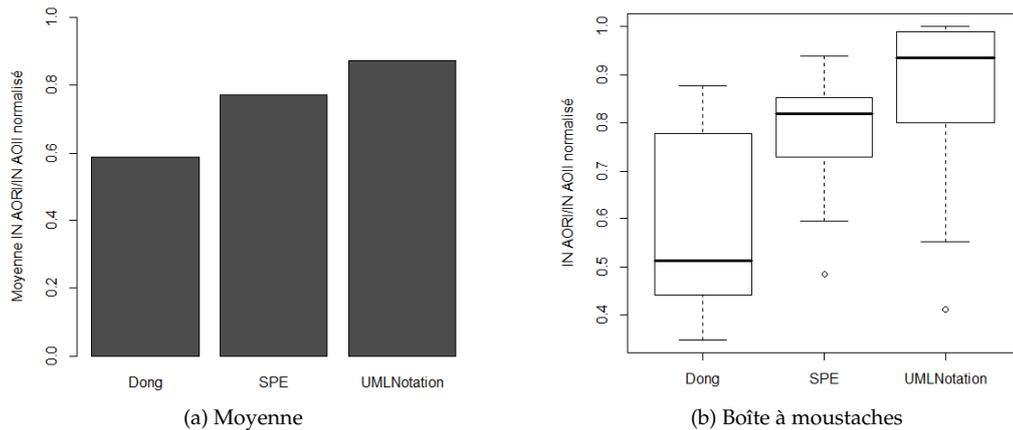


FIGURE 11.5 – Étude du taux IN AORI/IN AOII normalisé en fonction de la représentation - partie question

Tests d'hypothèses

Comme le lecteur peut le remarquer sur le graphique 11.4a, les sujets exposent, en moyenne, la densité de transition la plus faible lorsqu'ils utilisent UMLNotation. Ainsi, cette représentation favorise des transitions directes et un meilleur parcours visuel. Par ailleurs, Dong occasionne une densité de transition moyenne plus élevée que la nouvelle représentation proposée (SPE). La présence des couleurs dans cette dernière est probablement la cause de cette différence.

Le tableau 11.7 présente les p-values de comparaison des trois traitements. Deux valeurs se distinguent par un caractère légèrement significatif (< 0.1) et une valeur démontre une influence significative (< 0.05). Par conséquent, ces valeurs nous permettent de rejeter les trois hypothèses nulles (H_{0a} , H_{0b} et H_{0c}) pour un $\alpha = 0.1$ et d'affirmer (avec une probabilité d'erreur inférieure à 10%) :

Conclusion 23. *Pour la partie « question » de la tâche « participation », avec UMLNotation, les sujets ont, en moyenne, une densité de transition significativement plus faible qu'avec SPE ou Dong.*

Conclusion 24. *Pour la partie « question » de la tâche « participation », avec SPE, les sujets ont, en moyenne, une densité de transition significativement plus faible qu'avec Dong.*

11.3.3 Étude de l'influence de la représentation : IN AORI/IN AOII normalisé - partie question

Données

Comme nous l'avons relevé dans la sous-section 11.1.2, le taux IN AORI/IN AOII normalisé est influencé par la représentation, tant dans la partie question que dans la partie « support » de la tâche. Par conséquent, nous allons étudier, dans cette sous-section l'influence pour la partie question. Les figures 11.5a et 11.5b se rapportent aux moyennes et à la distribution des données de la partie question. Les données utilisées dans cette sous-section sont contenues dans le tableau A.3.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
IN AORI/IN AOII	0.04281	0.01604	0.00009759

Tableau 11.8 – P-value du taux IN AORI/IN AOII normalisé pour la tâche « participation » en fonction de la représentation utilisée - partie question

Hypothèses

H_{0a}	Pour la partie « question » de la tâche, le taux IN AORI/IN AOII normalisé est, en moyenne, le même avec Dong et SPE.
H_{0b}	Pour la partie « question » de la tâche, le taux IN AORI/IN AOII normalisé est, en moyenne, le même avec UMLNotation et SPE.
H_{0c}	Pour la partie « question » de la tâche, le taux IN AORI/IN AOII normalisé est, en moyenne, le même avec Dong et UMLNotation.
H_{1a-1}	Pour la partie « question » de la tâche, avec Dong, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec SPE.
H_{1a-2}	Pour la partie « question » de la tâche, avec SPE, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec Dong.
H_{1b-1}	Pour la partie « question » de la tâche, avec UMLNotation, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec SPE.
H_{1b-2}	Pour la partie « question » de la tâche, avec SPE, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec UMLNotation.
H_{1c-1}	Pour la partie « question » de la tâche, avec Dong, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec UMLNotation.
H_{1c-2}	Pour la partie « question » de la tâche, avec UMLNotation, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec Dong.

Tests d'hypothèses

D'après la figure 11.5a, nous pouvons remarquer que les sujets ont un taux IN AORI/IN AOII normalisé plus élevé avec la représentation UMLNotation. Ainsi, les sujets ayant utilisé cette représentation se sont plus souvent concentrés sur des points pertinents que ceux ayant utilisé Dong ou SPE. Le graphique nous révèle également un taux supérieur pour SPE, s'il est comparé à Dong.

Le tableau 11.8 expose des résultats significatifs (avec $\alpha = 0.05$) pour le test d'hypothèse Mann-Whitney. Les trois hypothèses nulles peuvent donc être rejetées au profit des hypothèses alternatives H_{1a-2} , H_{1b-1} et H_{1c-2} . La différence significative entre le taux de Dong et le taux de SPE apporte une information intéressante sur SPE. Nous sommes tenté de penser que cette différence peut potentiellement trouver son explication dans les caractéristiques de SPE, à savoir l'information textuelle réduite et les couleurs. Dès lors, il semblerait que ces deux particularités amènent le sujet à se concentrer plus souvent sur des points pertinents.

Par conséquent nous pouvons affirmer les deux conclusions suivantes (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 25. *Pour la partie « question » de la tâche « participation », les sujets utilisant UML-Notation ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que les sujets utilisant SPE ou Dong.*

Conclusion 26. *Pour la partie « question » de la tâche « participation », les sujets utilisant SPE ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que les sujets utilisant Dong.*

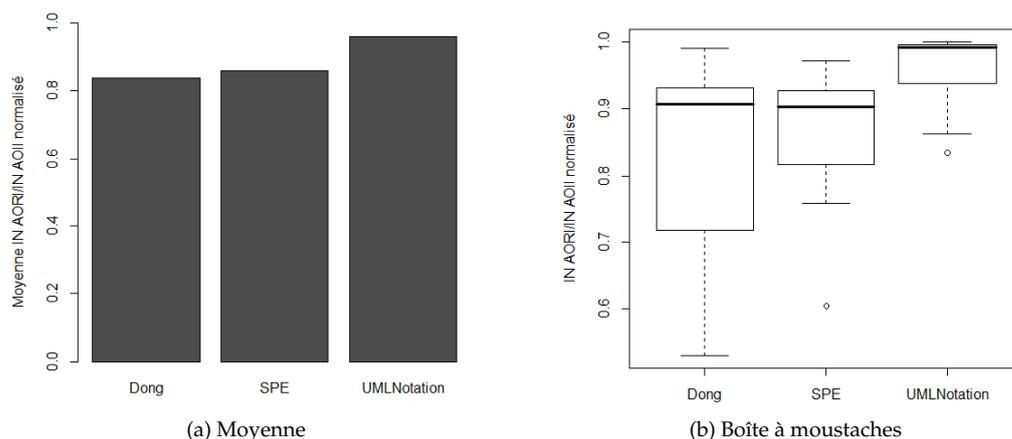


FIGURE 11.6 – Étude du taux IN AORI/IN AOII normalisé en fonction de la représentation - partie support

11.3.4 Étude de l'influence de la représentation : IN AORI/IN AOII normalisé - partie support

Données

Les figures 11.6a et 11.6b présentent respectivement les moyennes et la distribution du taux IN AORI/IN AOII normalisé par représentation pour la partie « support » de la tâche. Le tableau A.4 contient les données qui sont utilisées pour cette analyse.

Hypothèses

H_{0a}	Pour la partie « support » de la tâche, le taux IN AORI/IN AOII normalisé est, en moyenne, le même avec Dong et SPE.
H_{0b}	Pour la partie « support » de la tâche, le taux IN AORI/IN AOII normalisé est, en moyenne, le même avec UMLNotation et SPE.
H_{0c}	Pour la partie « support » de la tâche, le taux IN AORI/IN AOII normalisé est, en moyenne, le même avec Dong et UMLNotation.
H_{1a-1}	Pour la partie « support » de la tâche, avec Dong, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec SPE.
H_{1a-2}	Pour la partie « support » de la tâche, avec SPE, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec Dong.
H_{1b-1}	Pour la partie « support » de la tâche, avec UMLNotation, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec SPE.
H_{1b-2}	Pour la partie « support » de la tâche, avec SPE, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec UMLNotation.
H_{1c-1}	Pour la partie « support » de la tâche, avec Dong, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec UMLNotation.
H_{1c-2}	Pour la partie « support » de la tâche, avec UMLNotation, les sujets ont, en moyenne, un taux IN AORI/IN AOII plus élevé qu'avec Dong.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
IN AORI/IN AOII	0.9358	0.0009515	0.0008779

Tableau 11.9 – P-value du taux IN AORI/IN AOII normalisé pour la tâche « participation » en fonction de la représentation utilisée - partie support

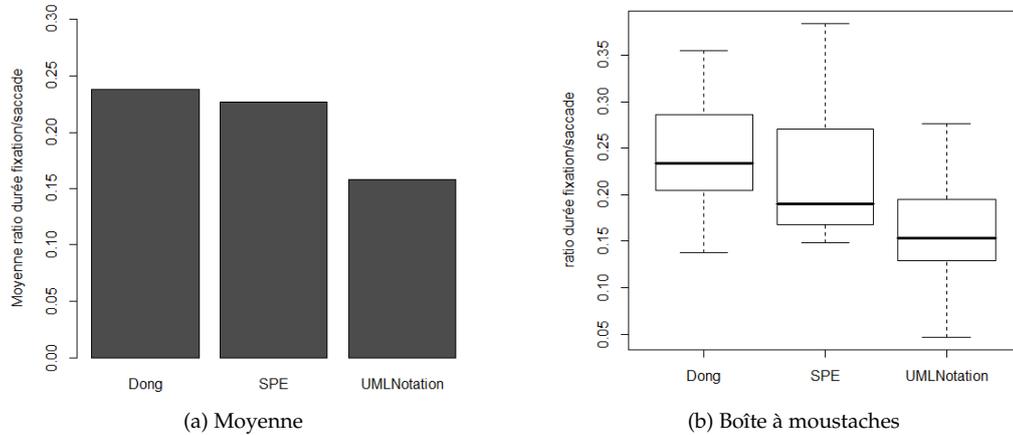


FIGURE 11.7 – Étude du ratio durée fixation/saccade en fonction de la représentation

Tests d'hypothèses

En raison du caractère non normal des distributions, le test Mann-Whitney a été utilisé pour calculer les p-values de comparaison des paires de traitements. Le tableau 11.9 expose celles-ci et démontre la présence de deux valeurs significatives pour un $\alpha = 0.05$: les comparaisons UMLNotation-SPE (0.0009515) et Dong-UMLNotation (0.0008779). Ainsi, en examinant le graphique des moyennes (figure 11.6a), nous constatons qu'UMLNotation possède le taux le plus élevé.

Plus formellement, nous pouvons réfuter les hypothèses H_{0b} et H_{0c} et tirer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 27. *Pour la partie « support » de la tâche « participation », les sujets utilisant UMLNotation ont, en moyenne, un taux IN AORI/IN AOII normalisé plus élevé que les sujets utilisant SPE ou Dong.*

11.3.5 Étude de l'influence de la représentation : ratio durée fixation/saccade

Données

Le graphique 11.7a illustre les moyennes du ratio durée fixation/saccade par représentation. La figure 11.7b présente les boîtes à moustaches des représentations pour la métrique étudiée. Les données utilisées dans cette sous-section sont accessibles dans le tableau A.3.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Ratio durée fixation/saccade	0.4115	0.01983	0.0006562

Tableau 11.10 – P-value du ratio durée fixation/saccade pour la partie « question » de la tâche « participation » en fonction de la représentation utilisée

Hypothèses

H_{0a}	Le ratio durée fixation/saccade est, en moyenne, le même avec Dong et SPE.
H_{0b}	Le ratio durée fixation/saccade est, en moyenne, le même avec UMLNotation et SPE.
H_{0c}	Le ratio durée fixation/saccade est, en moyenne, le même avec Dong et UML-Notation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant Dong.

Tests d'hypothèses

Le tableau 11.10 contient le résultat du test d'hypothèse Mann-Whitney utilisé pour la comparaison des paires de traitements. Comme la figure 11.7a le suggère, la représentation utilisée influence bien la métrique étudiée. Effectivement, en tenant compte des valeurs du tableau et d'un $\alpha = 0.05$, il existe une différence significative pour UMLNotation-SPE (0.01983) et pour Dong-UMLNotation (0.0006562). Par conséquent, les sujets ayant utilisés UMLNotation manifestent un ratio durée fixation/saccade significativement plus faible que ceux ayant utilisés les deux autres représentations. SPE possèdent également un rapport moyen légèrement plus faible que Dong, mais malheureusement, cette différence ne semble pas significative.

Dés lors, il est possible de rejeter les hypothèses nulles H_{0b} et H_{0c} , afin d'affirmer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 28. *Pour la partie « question » de la tâche « participation », les sujets utilisant UML-Notation ont, en moyenne, un ratio durée fixation/saccade plus faible que les sujets utilisant SPE ou Dong.*

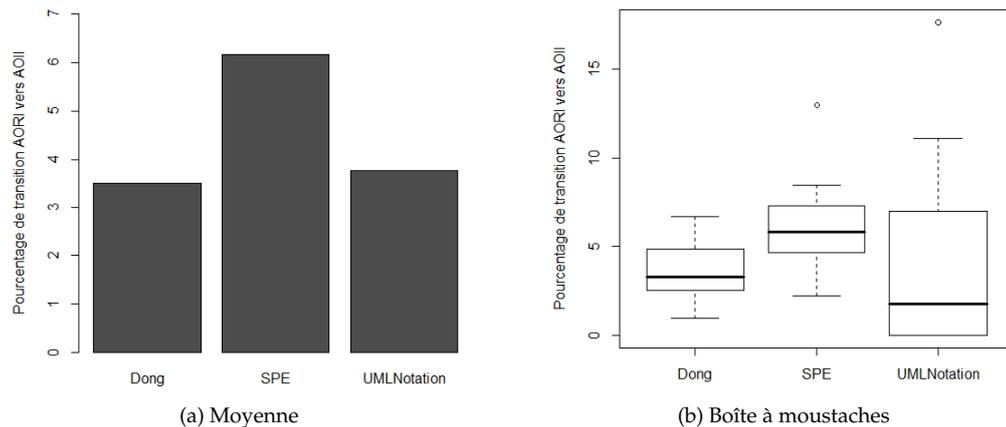


FIGURE 11.8 – Étude du pourcentage de transition AORI vers AOII en fonction de la représentation

11.3.6 Étude de l'influence de la représentation : pourcentage de transition AORI vers AOII

Données

Les différentes données utilisées pour cette analyse sont détaillées dans le tableau A.3. Le graphique 11.8a illustre les moyennes du pourcentage de transition AORI vers AOII en fonction de la représentation. La figure 11.8b présente les boîtes à moustaches des représentations pour la métrique étudiée.

Hypothèses

H_{0a}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même avec Dong et SPE.
H_{0b}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même avec UMLNotation et SPE.
H_{0c}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même avec Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant Dong.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Pourcentage de transition AORI vers AOII	0.02307	0.05616	0.3347

Tableau 11.11 – P-value du pourcentage de transition AORI vers AOII pour la partie « question » de la tâche « participation » en fonction de la représentation utilisée

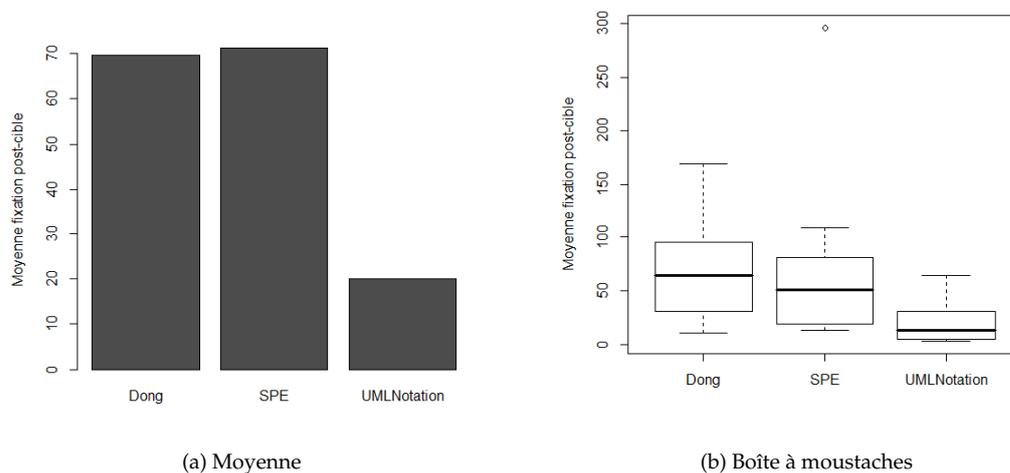


FIGURE 11.9 – Étude du nombre moyen de fixations pos-AORI en fonction de la représentation

Tests d'hypothèses

Le tableau 11.11 met en évidence les p-values du test Mann-Whitney utilisé pour la comparaison des paires de traitements. Selon ces valeurs, les différences Dong-SPE (0.02307) et UMLNotation-SPE (0.05616) sont significatives avec un $\alpha = 0.1$. La moyenne de la métrique étant la plus élevée avec SPE, nous pouvons réfuter les hypothèses nulles H_{0a} et H_{0b} , en faveur des hypothèses alternatives H_{1a-2} et H_{1b-2} . Comme nous l'avons déjà énoncé pour la tâche « composition » (section 10.3.3), le problème de différenciation des couleurs expose un pourcentage élevé de transition AORI vers AOII avec SPE.

Par conséquent, nous pouvons affirmer la conclusion suivante (avec une probabilité d'erreur inférieure à 10%) :

Conclusion 29. *Pour la partie « question » de la tâche « participation », les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que les sujets utilisant UMLNotation ou Dong.*

11.3.7 Étude de l'influence de la représentation : nombre moyen de fixations post-AORI

Données

Les différentes données utilisées pour cette analyse sont détaillées dans le tableau A.3. Le graphique 11.9a illustre les moyennes du nombre moyen de fixations post-AORI en fonction de la représentation. La figure 11.9b présente les boîtes à moustaches des représentations pour la métrique étudiée.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Nombre moyen de fixations post-AORI	0.6867	0.001565	0.0002389

Tableau 11.12 – P-value du nombre moyen de fixations post-AORI pour la partie « question » de la tâche « participation » en fonction de la représentation utilisée

Hypothèses

H_{0a}	Le nombre moyen de fixations post-AORI est, en moyenne, la même avec Dong et SPE.
H_{0b}	Le nombre moyen de fixations post-AORI est, en moyenne, la même avec UMLNotation et SPE.
H_{0c}	Le nombre moyen de fixations post-AORI est, en moyenne, la même avec Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que ceux utilisant Dong.

Tests d'hypothèses

La figure 11.9a illustre formellement une différence de moyenne suivant la représentation utilisée. Le tableau 11.12, contenant le résultat du test Mann-Whitney, confirme cette intuition en exposant deux valeurs significatives (avec un $\alpha = 0.05$) : UMLNotation vs SPE (0.001565) et Dong vs UMLNotation (0.0002389). Ainsi, Dong et SPE possèdent un nombre moyen de fixations post-AORI significativement plus élevé qu'UMLNotation. Un résultat élevé pour cette métrique décele un manque de visibilité ou un manque de mise en évidence des zones d'intérêts. Par ailleurs, aucune différence significative n'est observée pour la comparaison de Dong et de SPE.

Par conséquent, nous pouvons rejeter les hypothèses nulles H_{0b} et H_{0c} et affirmer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 30. *Pour la partie « question » de la tâche « participation », les sujets utilisant SPE ou Dong ont, en moyenne, un nombre moyen de fixations post-AORI plus élevé que les sujets utilisant UMLNotation.*

11.4 Analyse des distances relatives entre chemins visuels

Cette section a pour objectif de présenter en détail les résultats préalablement énoncés dans la sous-section 11.1.3. Comme nous l'avons remarqué dans le tableau 11.3, le sexe des participants n'in-

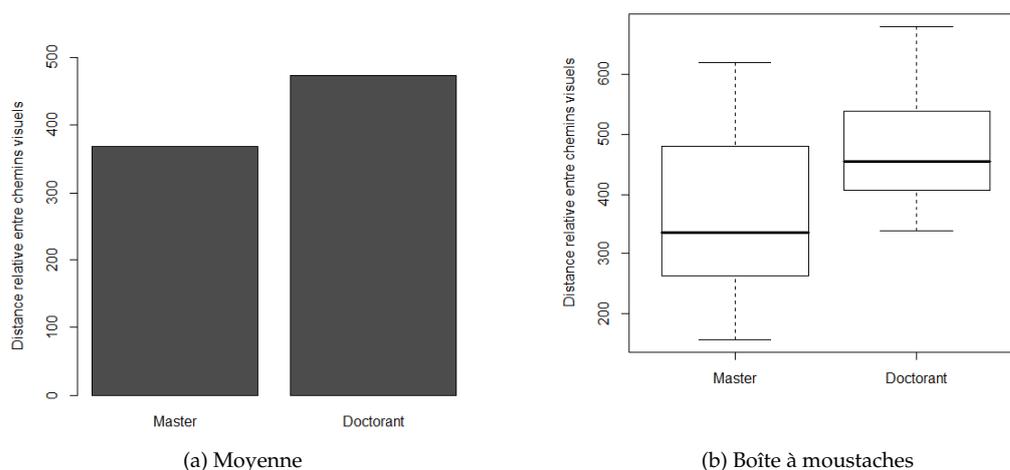


FIGURE 11.10 – Étude de la distance relative entre chemins visuels en fonction du niveau d'étude des participants pour la tâche « participation »

fluence en rien la métrique étudiée. Par conséquent, le niveau d'étude des participants est le seul facteur confondant pris en compte dans cette section.

11.4.1 Données

Les données utilisées pour cette analyse sont reprises en détails au sein du tableau A.6. Les moyennes des distances relatives entre chemins visuels en fonction du niveau d'étude sont illustrées par le graphique de la figure 11.10a. La figure 11.10b présente la répartition des distances également en fonction du niveau d'étude.

11.4.2 Hypothèses

H_0 La distance relative entre chemins visuels, est, en moyenne, la même quel que soit le niveau d'étude des participants.

H_1 La distance relative entre chemins visuels, est, en moyenne, plus faible pour le niveau d'étude X .

$X \in \{\text{Master, Doctorant}\}$

11.4.3 Tests d'hypothèses

Le test non-paramétrique Mann-Whitney expose une p-value égale à 0.01284. Concrètement, la figure 11.10a révèle une moyenne de distance relative entre chemins visuels plus élevée chez les doctorants. Le résultat du test d'hypothèse nous permet même d'annoncer que cette différence est significative.

Dès lors, la p-value résultante nous autorise à rejeter H_0 au profit de H_1 et à conclure (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 31. *Pour la partie « question » de la tâche « participation », les masters ont, en moyenne, une distance relative entre chemins visuels significativement plus faible que les doctorants.*

11.5 Conclusion de l'analyse de la tâche

Ce chapitre concentre l'ensemble des analyses portant sur la tâche de participation des classes, consistant à identifier toutes les classes participant à un patron de conception particulier. Ces analyses permettent de tirer plusieurs conclusions. D'une manière générale, aucune conclusion sur la non-influence d'une variable indépendante n'est apparue au cours des différentes sections.

Au sein de ce chapitre, trois conclusions ont relevé l'influence du framework utilisé sur certaines métriques de performance ou sur certaines métriques propres à l'oculométrie. L'analyse de la performance présente une influence du framework sur le temps nécessaire à la recherche des éléments de réponse de la tâche (conclusion 12). De plus, l'étude des mesures physiologiques démontre un impact du système sur deux métriques, en considérant la partie question et la partie « support » de la tâche. La première métrique apporte une deuxième conclusion relative au framework (conclusion 17). Celle-ci démontre un impact légèrement significatif ($\alpha = 0.1$) sur la métrique IN AORI/IN AOII normalisé, lorsque le sujet recherche les éléments de réponses. Enfin, la deuxième métrique amène la troisième et dernière conclusion. Celle-ci indique, lors de la partie support, un effet léger du framework sur la durée moyenne des fixations (conclusion 18).

Par ailleurs, les sections précédentes démontrent un impact de la représentation utilisée sur huit métriques. Pour la partie « support » de la tâche, seul la p-value du taux IN AORI/IN AOII normalisé était significative. L'analyse détaillée de cette influence conclut que les sujets, rassemblant les éléments de réponse, se concentrent plus souvent sur des points pertinents lorsqu'ils utilisent UMLNotation (conclusion 27).

D'autre part, plusieurs conclusions visent la partie « question » de la tâche « participation ». Les conclusions 22 et 23 démontrent, respectivement, une densité spatiale et une densité de transition significativement plus faible en cas d'utilisation d'UMLNotation. Le taux IN AORI/IN AOII normalisé expose également un taux supérieur chez les utilisateurs d'UMLNotation (conclusion 25). Par conséquent, les conclusions 22, 23 et 25 nous permettent d'affirmer que les sujets, utilisant UMLNotation, procèdent à des recherches directes et concentrées sur des zones pertinentes. De plus, il s'avère que les sujets utilisant UMLNotation possèdent un ratio durée fixation/saccade significativement plus faible que ceux utilisant les deux autres représentations (conclusion 28). Comme la recherche est directe et concentrée avec UMLNotation, il est probable que cette différence soit due à un temps de traitement plus faible. En outre, la comparaison de la densité de transition pour SPE et Dong affirme, quant à elle, une densité significativement plus faible pour la première représentation (conclusion 24). Ainsi, les conclusions 24 et 26 témoignent une recherche directe et une concentration plus régulière sur les zones d'intérêt de la part des utilisateurs de SPE. Comme nous l'avons déjà relevé pour la tâche « composition », le problème de différenciation des couleurs utilisées dans SPE est caractérisé par un pourcentage de transition AORI vers AOII élevé (conclusion 29). Enfin, suivant la conclusion 30, il semble que Dong et SPE souffrent d'un manque de visibilité ou d'un manque de mise en évidence des zones pertinentes.

Enfin, la conclusion 31 manifeste un impact du niveau d'étude sur la distance relative entre chemins visuels. Pour la tâche « participation », il semble que les masters ont une valeur, pour cette

métrique, significativement plus faible que les doctorants. Or, cette conclusion avait déjà été relevée pour la tâche de composition de patron de conception (chapitre 10).

Analyse de la tâche « rôle »

Ce chapitre contient l'analyse des résultats de la tâche d'identification de rôles. Au cours de cette tâche, le sujet doit identifier tous les rôles joués par une certaine classe. La structuration des analyses utilisée au sein de ce chapitre est la même que celle décrite en section 10.1.

La section 12.1 contient la sélection des métriques significatives des trois axes d'analyses utilisés. L'étude détaillée de l'influence de la représentation sur les métriques de performance est présentée dans la section 12.2. Ensuite, la section 12.3 approfondit l'analyse de l'impact des représentations sur les mesures oculométriques. L'analyse détaillée de la distance relative entre chemins visuels, suivant les facteurs confondants, est réalisée au sein de la section 12.4. Enfin la section 12.5 énonce les différentes conclusions relevées dans les sections de ce chapitre.

Sommaire

12.1	Sélection des métriques significatives	142
12.1.1	Sélection des métriques de performance significatives	142
12.1.2	Sélection des métriques physiologiques significatives	143
12.1.3	Sélection des distances relatives entre chemins visuels significatives	143
12.2	Analyse des performances	144
12.2.1	Étude de l'influence de la représentation : validité des réponses	144
12.3	Analyse des mesures physiologiques	145
12.3.1	Étude de l'influence de la représentation : durée moyenne des fixations	145
12.3.2	Étude de l'influence de la représentation : ratio durée fixation/saccade	147
12.3.3	Étude de l'influence de la représentation : pourcentage de transition AORI vers AOII	149
12.4	Distances relatives entre chemins visuels	150
12.4.1	Données	150
12.4.2	Hypothèses	151
12.4.3	Tests d'hypothèses	151
12.5	Conclusion de l'analyse de la tâche	151

	Framework	Représentation
Exactitude	1	0.07913
Temps	0.5703	0.1803

Tableau 12.1 – Résumé des tests d’hypothèses des mesures de performance en fonction des variables indépendantes pour la tâche « rôle »

Mesure	Question/Support	Framework	Représentation
Densité spatiale	Question	0.938	0.2104
	Support	0.4082	0.873
Densité de transition	Question	0.006007	0.9725
	Support	-	-
IN AORI/IN AOII normalisé	Question	0.8948	0.4999
	Support	0.281	0.9678
Durée moyenne de fixation	Question	0.93058	0.00269
	Support	0.3468	0.933
Ratio durée fixation/saccade	Question	0.21880	0.00279
	Support	-	-
Pourcentage de transition AORI vers AOII	Question	0.1265	0.005599
	Support	-	-
Nombre moyen de fixations post-AORI	Question	0.913	0.1542
	Support	-	-

Tableau 12.2 – Résumé des p-values des mesures physiologiques en fonction des variables indépendantes pour la tâche « rôle »

12.1 Sélection des métriques significatives

12.1.1 Sélection des métriques de performance significatives

Cette sous-section présente le résultat des tests d’hypothèses réalisés sur les mesures de performance, à savoir le temps passé sur une tâche et l’exactitude des réponses. Les p-values résultantes des tests d’hypothèses appliquées aux métriques de performance sont présentées dans le tableau 12.1. Les tests Mann-Whitney et Kruskal-Wallis ont été utilisés, respectivement, pour vérifier l’influence des frameworks et des représentations. Le lecteur remarquera directement que seul la p-value d’exactitude des réponses (0.07913), suivant la représentation, est significative pour un $\alpha = 0.1$. Une étude détaillée de cette influence sera réalisée en section 12.2.

Par ailleurs, le tableau présente une valeur qui détermine une non-influence du framework sur l’exactitude des réponses ($p - value = 1$). En effet, une valeur, calculée à partir de l’unité à laquelle on soustrait la p-value (1-pvalue), inférieure à α nous autorise à énoncer cette conclusion.

Ces deux observations nous permettent de tirer deux conclusions :

Conclusion 32. *Pour la tâche « rôle », l’exactitude des réponses n’est pas influencée par le framework utilisé.*

Conclusion 33. *Pour la tâche « rôle », l’exactitude des réponses est légèrement influencée par la représentation utilisée.*

12.1.2 Sélection des métriques physiologiques significatives

Le résumé des tests d'hypothèses accompli sur les mesures oculométriques est présenté dans cette sous-section. Cette analyse tient compte des mêmes mesures que celles employées dans les sections 10.2.2 et 11.1.2.

Le tableau 12.2 expose les p-values d'influence du framework et de la représentation. En raison du caractère non normal des distributions, ces valeurs ont été calculées à partir des tests Mann-Whitney (pour le framework) et Kruskal-Wallis (pour la représentation).

La partie du tableau se rapportant à l'influence du framework contient une seule valeur significative (pour un $\alpha = 0.05$). L'influence sur la densité de transition est donc la seule conclusion que nous pouvons tirer concernant cette variable indépendante.

D'un autre côté, les p-values qui ont un rapport avec la représentation indiquent une influence de celle-ci sur trois métriques. Les p-values de la durée moyenne des fixations, du ratio durée fixation/saccade et du pourcentage de transition AORI vers AOII sont inférieures à 0.05 et sont donc significatives. Le tableau contient également deux valeurs exprimant la non-influence de la variable indépendante sur les métriques : densité de transition (partie question) et IN AORI/IN AOII normalisé (partie support).

Plusieurs conclusions peuvent donc être tirées :

Conclusion 34. *Pour la partie « question » de la tâche « rôle », la densité de transition n'est pas influencée par la représentation utilisée.*

Conclusion 35. *Pour la partie « support » de la tâche « rôle », le taux IN AORI/IN AOII normalisé n'est pas influencé par la représentation utilisée.*

Conclusion 36. *Pour la partie « question » de la tâche « rôle », les variables : durée moyenne de fixation, ratio durée fixation/saccade et pourcentage de transition AORI vers AOII sont influencées par la représentation utilisée.*

Conclusion 37. *Pour la partie « question » de la tâche « rôle », la densité de transition est influencée par le framework utilisé.*

12.1.3 Sélection des distances relatives entre chemins visuels significatives

Le tableau 12.3 inclut les résultats des tests d'hypothèses des deux groupes sur la métrique étudiée. La valeur se rapportant au sexe provient du test Mann-Whitney. Tandis que la valeur du niveau d'étude a été calculée à partir du test ANOVA (car les données suivent une distribution normale). Si nous considérons un $\alpha = 0.1$, les p-values du tableau démontrent une influence légèrement significative du sexe des participants sur la métrique étudiée.

L'approfondissement de cette influence sera présenté en section 12.4. Dans la sous-section courante, nous nous limiterons à accepter la conclusion suivante :

Conclusion 38. *Pour la tâche « rôle », la distance relative entre chemins visuels est légèrement influencée par le sexe des sujets.*

	Sexe	Niveau d'étude
DRCV	0.05784	0.8083

Tableau 12.3 – Résumé des tests d'hypothèses sur la mesure de la distance relative entre les chemins visuels en fonction des facteurs confondants pour la tâche « rôle »

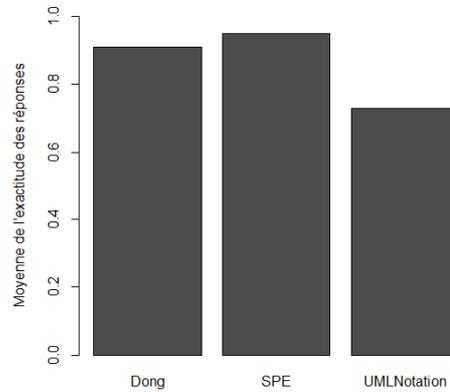


FIGURE 12.1 – Étude de l'exactitude des réponses de la tâche « rôle » en fonction de la représentation

12.2 Analyse des performances

Cette section présente l'analyse approfondie de la conclusion 33, c'est-à-dire l'étude de l'influence des représentations sur le taux d'exactitude des réponses pour la tâche « rôle ».

12.2.1 Étude de l'influence de la représentation : validité des réponses

Données

La figure 12.1 présente, au lecteur, les moyennes des taux de bonnes réponses en fonction des représentations. Les données utilisées proviennent du tableau A.2.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Exactitude des réponses	0.6314	0.05658	0.1223

Tableau 12.4 – P-value du caractère exact des réponses de la tâche « rôle » en fonction de la représentation utilisée

Hypothèses

H_{0a}	Le taux de réponses exactes est, en moyenne, le même, pour les sujets utilisant Dong et SPE.
H_{0b}	Le taux de réponses exactes est, en moyenne, le même, pour les sujets utilisant UMLNotation et SPE.
H_{0c}	Le taux de réponses exactes est, en moyenne, le même, pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant Dong.

Tests d'hypothèses

Suivant les moyennes illustrées par le graphique 12.1, les sujets ont un meilleur taux de bonnes réponses lorsqu'ils utilisent SPE. Nous remarquons également que la p-value de comparaison UMLNotation-SPE du tableau 11.1 (0.05658) est significative pour un $\alpha = 0.1$. Nous pouvons donc rejeter H_{0b} et valider l'hypothèse H_{1b-2} (avec une probabilité d'erreur inférieure à 10%) :

Conclusion 39. *Pour la partie « question » de la tâche « rôle », les sujets utilisant SPE possèdent, en moyenne, un taux de réponses exactes significativement plus élevé que ceux utilisant UMLNotation.*

12.3 Analyse des mesures physiologiques

Cette section contient l'étude approfondie de la conclusion 36. Par ailleurs, la conclusion propre à l'influence du framework (conclusion 37) ne sera pas examinée en détail.

12.3.1 Étude de l'influence de la représentation : durée moyenne des fixations

Données

Les graphiques 12.2a et 12.2b illustrent, respectivement, les moyennes et les boîtes à moustaches de la durée moyenne des fixations suivant la représentation utilisée. Ces deux figures présentent les données contenues dans le tableau A.3.

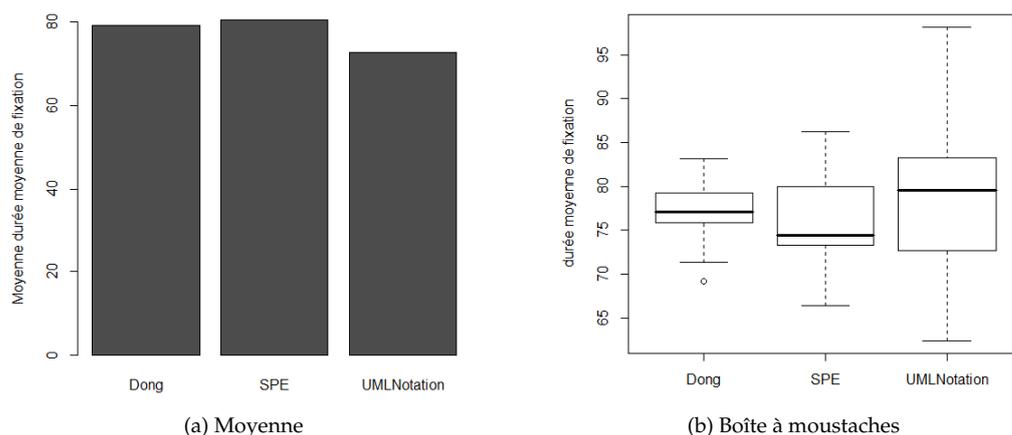


FIGURE 12.2 – Étude de la durée moyenne des fixations en fonction de la représentation

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Durée moyenne de fixations	0.8441	0.002767	0.003476

Tableau 12.5 – P-value de la durée moyenne des fixations pour la partie « question » de la tâche « rôle » en fonction de la représentation utilisée

Hypothèses

H_{0a}	La durée moyenne des fixations est, en moyenne, la même, pour les sujets utilisant Dong et SPE.
H_{0b}	La durée moyenne des fixations est, en moyenne, la même, pour les sujets utilisant UMLNotation et SPE.
H_{0c}	La durée moyenne des fixations est, en moyenne, la même, pour les sujets utilisant Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong possèdent, en moyenne, une durée moyenne des fixations significativement plus faible que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE possèdent, en moyenne, une durée moyenne des fixations significativement plus faible que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation possèdent, en moyenne, une durée moyenne des fixations significativement plus faible que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE possèdent, en moyenne, une durée moyenne des fixations significativement plus faible que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong possèdent, en moyenne, une durée moyenne des fixations significativement plus faible que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation possèdent, en moyenne, une durée moyenne des fixations significativement plus faible que ceux utilisant Dong.

Tests d'hypothèses

Comme le lecteur peut le remarquer sur le graphique des moyennes, les sujets détiennent la durée moyenne des fixations la plus faible lorsqu'ils utilisent UMLNotation. Cette faible valeur signifie qu'UMLNotation présente des informations qui sont rapidement traitées et interprétées par les sujets. SPE, par contre, possèdent la valeur la plus élevée.

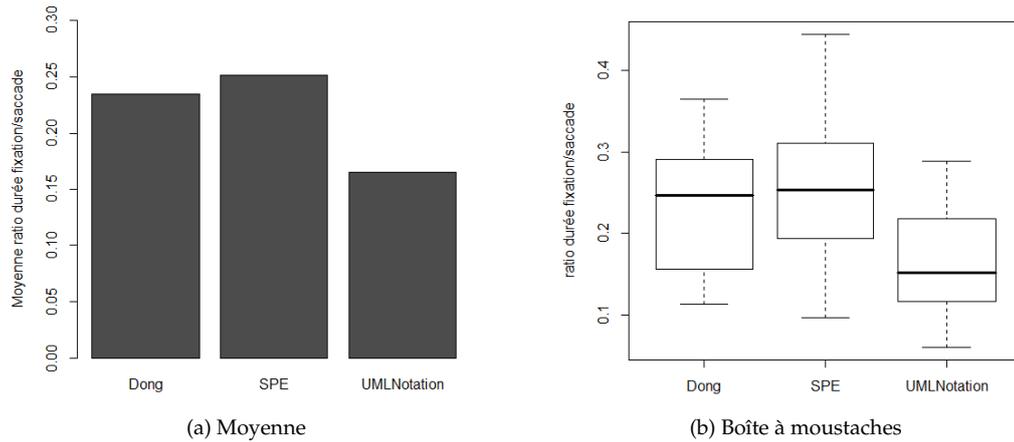


FIGURE 12.3 – Étude du ratio durée fixation/saccade en fonction de la représentation

Le tableau 12.5 présente les p-values de comparaison des trois traitements. Deux valeurs démontrent une influence significative (avec un $\alpha = 0.05$). Par conséquent, ces valeurs nous permettent de rejeter les hypothèses nulles H_{0b} et H_{0c} . Étonnamment, aucune différence significative n'est relevée pour la comparaison Dong-SPE. Malheureusement, aucune conclusion ne peut donc être énoncée concernant les avantages de la réduction de l'information textuelle présente chez SPE.

Plus formellement, il est possible de tirer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 40. *Pour la partie « question » de la tâche « rôle », avec UMLNotation, les sujets ont, en moyenne, une durée moyenne des fixations significativement plus faible qu'avec SPE ou Dong.*

12.3.2 Étude de l'influence de la représentation : ratio durée fixation/saccade

Données

Le graphique 11.7a illustre les moyennes du ratio durée fixation/saccade par représentation. La figure 11.7b présente les boîtes à moustaches des représentations pour la métrique étudiée. Les données utilisées dans cette sous-section sont accessibles dans le tableau A.3.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Ratio durée fixation/saccade	0.653	0.002282	0.006989

Tableau 12.6 – P-value du ratio durée fixation/saccade pour la partie « question » de la tâche « rôle » en fonction de la représentation utilisée

Hypothèses

H_{0a}	Le ratio durée fixation/saccade est, en moyenne, le même avec Dong et SPE.
H_{0b}	Le ratio durée fixation/saccade est, en moyenne, le même avec UMLNotation et SPE.
H_{0c}	Le ratio durée fixation/saccade est, en moyenne, le même avec Dong et UML-Notation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus élevé que ceux utilisant Dong.

Tests d'hypothèses

Le tableau 12.6 contient le résultat du test d'hypothèse Mann-Whitney utilisé pour la comparaison des paires de traitements. Comme la figure 11.7a le suggère, la représentation utilisée influence bien la métrique étudiée. Effectivement, en tenant compte des valeurs du tableau et d'un $\alpha = 0.05$, il existe une différence significative pour UMLNotation-SPE (0.002282) et pour Dong-UMLNotation (0.006989). Par conséquent, les sujets ayant utilisés UMLNotation manifestent un ratio durée fixation/saccade significativement plus faible que ceux ayant utilisés les deux autres représentations. Dong possèdent également un rapport moyen légèrement plus faible que SPE, mais malheureusement, cette différence ne semble pas significative.

Dès lors, il est possible de rejeter les hypothèses nulles H_{0b} et H_{0c} , afin d'affirmer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 41. *Pour la partie « question » de la tâche « rôle », les sujets utilisant UMLNotation ont, en moyenne, un ratio durée fixation/saccade plus faible que les sujets utilisant SPE ou Dong.*

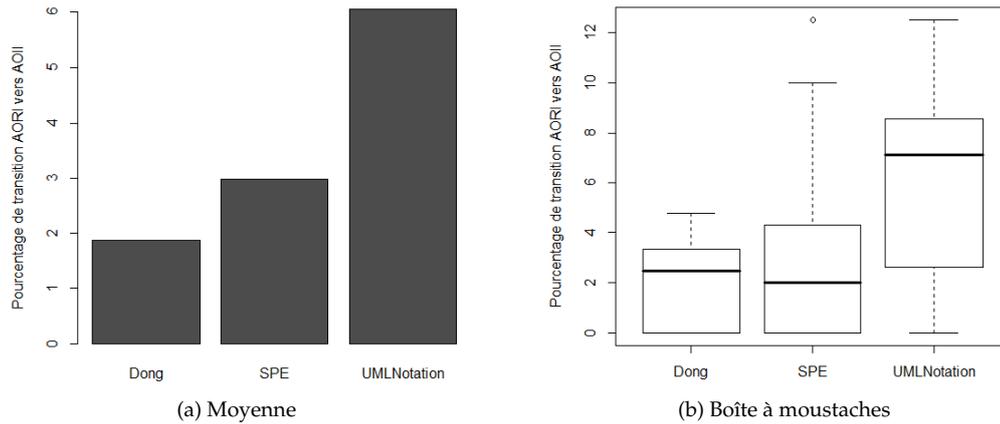


FIGURE 12.4 – Étude du pourcentage de transition AORI vers AOII en fonction de la représentation

12.3.3 Étude de l'influence de la représentation : pourcentage de transition AORI vers AOII

Données

Les différentes données utilisées pour cette analyse sont détaillées dans le tableau A.3. Le graphique 12.4a présente les moyennes du pourcentage de transition AORI vers AOII en fonction de la représentation. La figure 12.4b illustre les boîtes à moustaches des représentations pour la métrique étudiée.

Hypothèses

H_{0a}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même avec Dong et SPE.
H_{0b}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même avec UMLNotation et SPE.
H_{0c}	Le pourcentage de transition AORI vers AOII est, en moyenne, le même avec Dong et UMLNotation.
H_{1a-1}	Les sujets utilisant Dong ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant SPE.
H_{1a-2}	Les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant Dong.
H_{1b-1}	Les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant SPE.
H_{1b-2}	Les sujets utilisant SPE ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant UMLNotation.
H_{1c-1}	Les sujets utilisant Dong ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant UMLNotation.
H_{1c-2}	Les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que ceux utilisant Dong.

	Dong vs SPE	UMLNotation vs SPE	Dong vs UMLNotation
Pourcentage transition AORI vers AOII	0.6126	0.02663	0.001548

Tableau 12.7 – P-value du pourcentage de transition AORI vers AOII pour la partie « question » de la tâche « rôle » en fonction de la représentation utilisée

Tests d'hypothèses

Le tableau 12.7 met en évidence les p-values du test Mann-Whitney utilisé pour la comparaison des paires de traitements. Selon ces valeurs, les différences UMLNotation-SPE (0.02663) et Dong-UMLNotation (0.001548) sont significatives pour un $\alpha = 0.05$. La moyenne de la métrique étant la plus élevée avec UMLNotation, nous pouvons réfuter les hypothèses nulles H_{0b} et H_{0c} , en faveur des hypothèses alternatives H_{1b-1} et H_{1c-2} .

Suivant ces analyses, il semble qu'UMLNotation souffre d'une mauvaise interprétation des zones d'intérêt, lorsque les sujets recherchent les rôles joués par une classe. Cette constatation avait déjà été relevée pour la tâche « composition ». Par ailleurs, pour la tâche « rôle », la différence est présente lors de la comparaison avec les deux autres représentations. Par conséquent, les résultats donnent l'impression que le mélange des informations des rôles des patrons de conception avec les autres éléments du diagramme de classes porte préjudice à UMLNotation.

À partir de ces informations, nous pouvons affirmer la conclusion suivante (avec une probabilité d'erreur inférieure à 5%) :

Conclusion 42. *Pour la partie « question » de la tâche de « rôle », les sujets utilisant UMLNotation ont, en moyenne, un pourcentage de transition AORI vers AOII plus élevé que les sujets utilisant SPE ou Dong.*

12.4 Distances relatives entre chemins visuels

Cette section a pour objectif d'étudier en détail la conclusion 38 énoncée en section 12.1.3. Comme nous l'avons remarqué, le niveau d'étude des participants n'influence en rien la métrique étudiée pour la tâche « rôle ». Par conséquent, le sexe des sujets sera le seul facteur confondant pris en compte dans cette section.

12.4.1 Données

Les données utilisées pour cette analyse sont représentées en détails au sein du tableau A.5. Les moyennes des distances relatives entre chemins visuels en fonction du sexe des sujets sont illustrées par le graphique de la figure 12.5a. La figure 12.5b présente la répartition des distances également en fonction du sexe des participants.

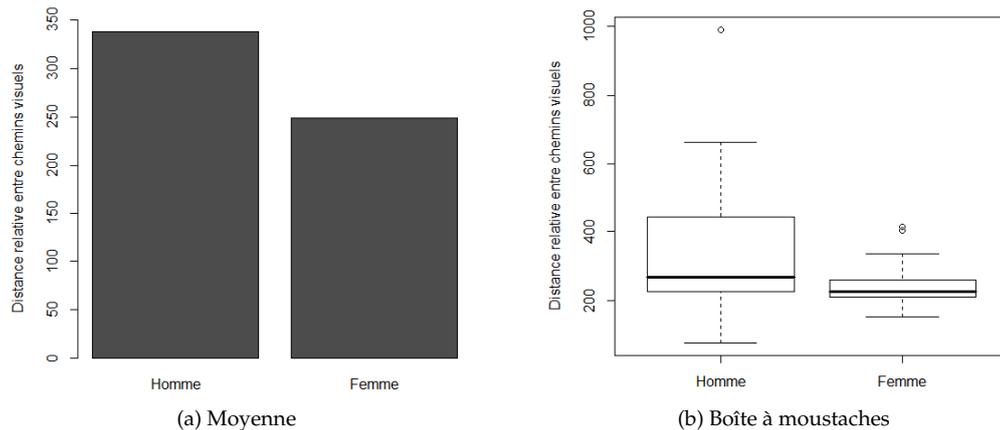


FIGURE 12.5 – Étude de la distance relative entre chemins visuels en fonction du sexe des participants pour la tâche « rôle »

12.4.2 Hypothèses

H_0 La distance relative entre chemins visuels, est, en moyenne, la même quel que soit le sexe des participants.

H_1 La distance relative entre chemins visuels, est, en moyenne, plus faible lorsque les participants sont du sexe X .

$X \in \{\text{Masculin, Féminin}\}$

12.4.3 Tests d'hypothèses

Pour tester l'hypothèse nulle, le test non-paramétrique Mann-Whitney a été utilisé. Concrètement, la figure 12.5a révèle que les hommes présentent une moyenne de distance relative entre chemins visuels plus élevée. Le résultat du test d'hypothèse ($= 0.05784$) nous permet même de dire que cette influence est significative (avec $\alpha = 0.1$). La boîte à moustaches contenue dans la figure 12.5b confirme cette observation. En effet, les distances relatives entre chemins visuels chez les hommes sont plus dispersées que chez les femmes. Ceci montre qu'elles semblent se comporter de façon plus homogène que les hommes. La p-value résultante nous autorise à rejeter H_0 , au profit de H_1 , et à conclure (avec une probabilité d'erreur inférieure à 10%) :

Conclusion 43. *Pour la partie « question » de la tâche « rôle », les femmes ont, en moyenne, une distance relative entre chemins visuels significativement plus faible que les hommes.*

12.5 Conclusion de l'analyse de la tâche

Ce chapitre contient l'ensemble des analyses portant sur la tâche consistant à identifier tous les rôles joués par une classe déterminée. Ces analyses nous autorisent à tirer plusieurs conclusions.

D'une part, l'analyse de la première variable indépendante (framework) permet de déduire deux conclusions. La première conclusion détermine une non-influence du framework sur le taux de validité des réponses, pour la tâche « rôle » (conclusion 32). La seconde conclusion concerne une

variable dépendante étudiée dans l'axe d'analyse physiologique. Celle-ci démontre un impact du framework sur la densité de transition (conclusion 37).

D'autre part, les sections précédentes annoncent la présence d'une influence de la représentation utilisée sur quatre métriques. Pour la partie « support » de la tâche, seule la p-value du taux IN AORI/IN AOII normalisé permet d'énoncer une conclusion. Cette valeur révèle une non-influence des représentations sur cette métrique (conclusion 35). Par ailleurs, cinq conclusions se rapportent à la partie « question » de la tâche. Tout d'abord, nous avons relevé une non-influence de la représentation sur la densité de transition (conclusion 34). Ensuite, les conclusions 40, 41 et 42 présentent une différence significative lorsque l'on compare UMLNotation à SPE et Dong. Premièrement, les utilisateurs d'UMLNotation ont une durée moyenne des fixations significativement plus faible que les utilisateurs de SPE ou Dong (conclusion 40). Cette faible valeur signifie qu'UMLNotation présente les informations des patrons d'une telle façon qu'elles peuvent être rapidement traitées et interprétées par les sujets. Ensuite, les sujets utilisant UMLNotation possèdent le ratio durée fixation/saccade le plus faible (conclusion 41). Ainsi, d'après ces deux conclusions, il semble que la durée du traitement est inférieure chez UMLNotation. En troisième lieu, avec UMLNotation, les sujets possèdent un pourcentage de transition des zones d'intérêt vers les zones de non-intérêt le plus élevé (conclusion 42). Comme cela avait déjà été relevé pour la tâche « composition », cette conclusion émet l'hypothèse qu'UMLNotation souffre d'une mauvaise interprétation de ses zones pertinentes. Enfin la dernière conclusion démontre qu'avec SPE, les sujets ont un taux de réponses exactes significativement plus élevé qu'avec UMLNotation (conclusion 39). Par conséquent, les conclusions 39 et 42 présentent le principal désavantage d'UMLNotation. En réalité, les sujets ont des problèmes pour différencier les éléments du diagramme de classes et les éléments des patrons de conception.

Pour finir, la conclusion 43 manifeste un impact du sexe des sujets sur la distance relative entre chemins visuels. Les résultats démontrent que les femmes ont, en moyenne, une distance relative entre chemins visuels significativement plus faible que les hommes. Les femmes semblent se comporter de façon plus homogènes que les hommes, lorsqu'elles identifient les rôles joués par une classe déterminée.

Analyse des mesures subjectives

La dernière partie de l'analyse de nos résultats concerne l'étude de l'évaluation subjective de la charge mentale ressentie par les sujets lors des tâches « composition », « participation » et « rôle ». L'évaluation subjective de cette charge mentale a été réalisée à partir du système de mesure NASA-TLX (cf. section 5).

La section 13.1 détaille l'approche qui est utilisée dans cette dernière partie de l'analyse des résultats. Ensuite, l'étude de l'influence des variables indépendantes et des facteurs secondaires (sexe et niveau d'étude) sont présentés, respectivement, en section 13.2 et en section 13.3. Enfin, la section 12.5 énonce les différentes conclusions relevées dans les sections de ce chapitre.

Sommaire

13.1	Approche et mise en place	154
13.2	Étude de l'influence des variables indépendantes	154
13.2.1	Étude de l'influence de la représentation	155
13.2.2	Étude de l'influence du framework	155
13.3	Étude de l'influence des facteurs confondants	156
13.3.1	Étude de l'influence du sexe	156
13.3.2	Étude de l'influence du niveau d'étude	157
13.4	Conclusions	158

Sous-échelle	Poids
Demande mentale	5
Demande physique	0
Demande temporelle	3
Performance	4
Effort	2
Frustration	1

Tableau 13.1 – Poids attribué à chaque sous-échelle de NASA-TLX

13.1 Approche et mise en place

Avant de présenter les différentes études, il est important de décrire la configuration du système NASA-TLX qui a été utilisée. Les données de cette métrique sont enregistrées grâce à un questionnaire. Comme expliqué dans l'exécution de l'expérience (section 9.2.3), chaque sujet remplit ce questionnaire dès qu'il a accompli les trois tâches (« composition », « participation » et « rôle ») se rapportant à un couple (framework, représentation). Ainsi, il est possible de comparer l'évaluation de la charge mentale de chaque sujet pour chaque couple de variable indépendante qui lui a été attribué.

Pour que NASA-TLX soit bel et bien une métrique représentative de la charge mentale, il est nécessaire de fixer des poids pour chaque sous-échelle. La détermination de ces poids dépend, essentiellement, de la tâche effectuée par les sujets. Le poids de chaque sous-échelle doit être compris dans l'intervalle [0..5]. Le tableau 13.1 contient notre attribution et sa justification est décrite ci-dessous.

Demande mentale : Dans notre attribution, la demande mentale possède le poids le plus élevé. En réalité, nous estimons que cette sous-échelle est la plus représentative de la charge mentale.

Demande physique : Notre expérience ne demandant pas d'effort physique au sujet, nous avons décidé d'attribuer un poids nul à la demande physique.

Demande temporelle : Le temps nécessaire au sujet pour réaliser les trois tâches est une donnée importante. La valeur moyenne attribuée est justifiée par le fait que nous n'avons pas effectué de pression temporelle sur le sujet.

Performance : Un poids de valeur 2 montre une importance de l'évaluation de la performance, tout en laissant de la puissance aux autres sous-échelles.

Effort : Un poids élevé est attribué à cette sous-échelle, car elle caractérise le sentiment d'effort éprouvé par le sujet lors de la réalisation des tâches.

Frustration : La frustration est une mesure qui semble peut pertinente, ainsi un poids très faible lui a été attribué.

Une fois ces poids sélectionnés, il est possible de déterminer la charge mentale d'un sujet à partir de la formule suivante :

$$CM = (DM * PDM) + (DP * PDP) + (DT * PDT) + (P * PP) + (E * PE) + (F * PF)$$

Le tableau A.7 contient l'évaluation de la charge mentale pour chaque sujet et sera donc utilisé comme source pour toutes les analyses contenues dans ce chapitre.

13.2 Étude de l'influence des variables indépendantes

Cette section contient l'étude de l'influence des deux variables indépendantes, le framework et la représentation, sur le niveau de charge mentale subjective.

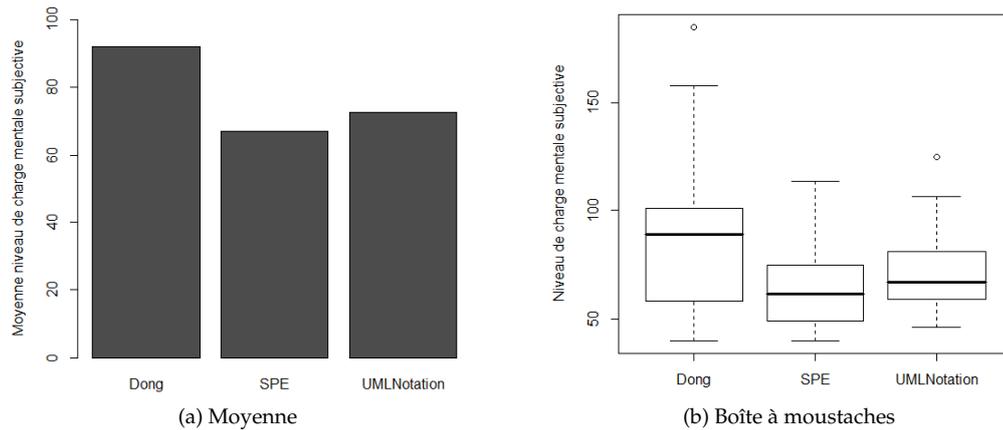


FIGURE 13.1 – Étude de la charge mentale subjective en fonction de la représentation

13.2.1 Étude de l'influence de la représentation

Données

Le graphique 13.1a illustre les moyennes des charges mentales subjectives par variante de représentation. Les boîtes à moustaches de ces données sont contenues dans la figure 13.2b.

Hypothèses

H_0 Les tâches de compréhension de patron de conception nécessitent, en moyenne, le même niveau de charge mentale subjective quel que soit la variante X de la représentation utilisée.

H_1 Avec la variante X , les sujets ont, en moyenne, un niveau de charge mentale subjective significativement plus faible pour les tâches de compréhension de patron de conception.

$$X \in \{\text{Dong, SPE, UMLNotation}\}$$

Test d'hypothèse

Comme les données suivent une distribution normale, il est possible d'utiliser le test paramétrique ANOVA. Le résultat de ce test est égal à 0.186. Ainsi, les différences présentées au sein du graphique 13.1a ne sont pas significatives pour un $\alpha = 0.05$. Suivant cette valeur, aucune conclusion ne peut être rejetée ou acceptée.

13.2.2 Étude de l'influence du framework

Données

Le graphique 13.2a illustre les moyennes des charges mentales subjectives en fonction des frameworks. Les boîtes à moustaches de ces données sont contenues dans la figure 13.2b.

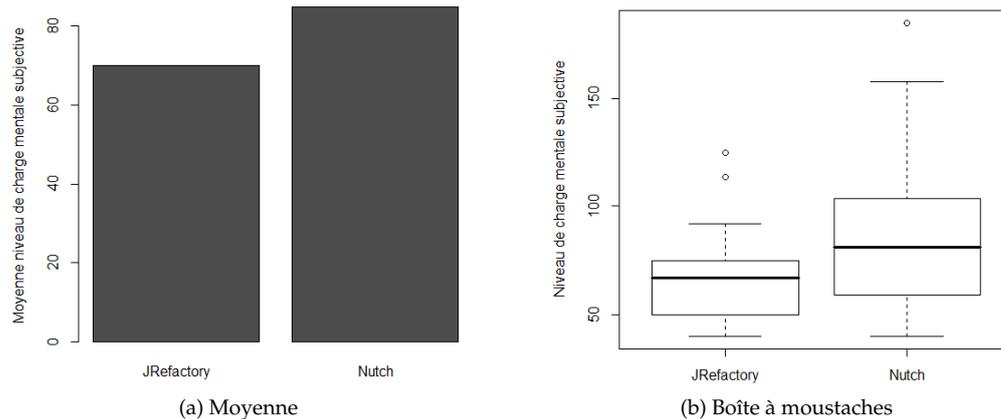


FIGURE 13.2 – Étude de la charge mentale subjective en fonction du framework utilisé

Hypothèses

H_0 Les tâches de compréhension de patron de conception nécessitent, en moyenne, le même niveau de charge mentale subjective quel que soit la variante Y du framework utilisé.

H_1 Avec la variante Y , les sujets ont, en moyenne, un niveau de charge mentale subjective significativement plus faible pour les tâches de compréhension de patron de conception.

$$Y \in \{\text{Nutch}, \text{JRefactory}\}$$

Test d'hypothèse

Le graphique 13.2a expose une différence entre les deux frameworks utilisés, avec une charge mentale plus élevée pour le framework Nutch. Cette observation correspond bien à la réalité, car le diagramme de classes de Nutch comporte beaucoup plus de liens que JRefactory. Ce phénomène diminue donc la lisibilité du diagramme et affecte la charge mentale. Malheureusement, d'après le résultat du test d'hypothèse Mann-Whitney (0.3702), la différence observée n'est pas significative (avec un $\alpha = 0.05$). Par conséquent, nous ne pouvons accepter ou refuser aucune hypothèse concernant l'influence du framework sur l'évaluation subjective de la charge mentale.

13.3 Étude de l'influence des facteurs confondants

Dans cette section, nous présentons les résultats de l'analyse de l'influence du sexe et du niveau d'étude sur l'évaluation subjective de la charge mentale.

13.3.1 Étude de l'influence du sexe

Données

Le graphique 13.3a illustre les moyennes des charges mentales subjectives en fonction du sexe des sujets. Les boîtes à moustaches de ces données sont contenues dans la figure 13.3b.

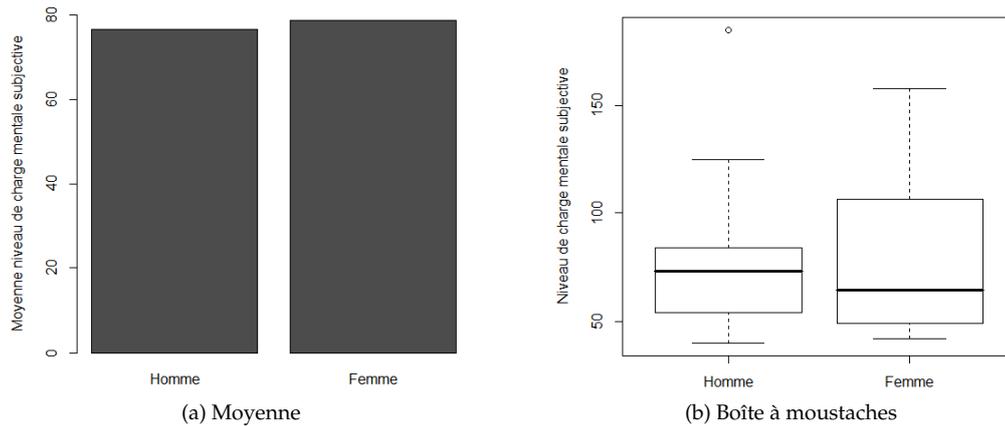


FIGURE 13.3 – Étude de la charge mentale subjective en fonction du sexe des sujets

Hypothèses

- H_0 Les tâches de compréhension de patron de conception nécessitent, en moyenne, le même niveau de charge mentale subjective quel que soit le sexe du sujet.
- H_1 Le niveau de charge mentale subjective est, en moyenne, significativement plus faible pour un sexe que pour l'autre.

Test d'hypothèse

Le lecteur remarquera directement, sur le graphique 13.3a, que la différence entre les deux distributions est très légère. En réalité, le résultat du test Mann-Whitney (0.9282) démontre que la différence qui existe n'est pas significative (pour un $\alpha = 0.05$). C'est pourquoi, il est impossible d'accepter ou de réfuter une hypothèse concernant l'influence du sexe des sujets sur l'évaluation subjective de la charge mentale.

13.3.2 Étude de l'influence du niveau d'étude

Données

Les graphiques 13.4a et 13.4b illustrent, respectivement, les moyennes et les boîtes à moustaches des charges mentales subjectives suivant le niveau d'étude des sujets.

Hypothèses

- H_0 Les tâches de compréhension de patron de conception nécessitent, en moyenne, le même niveau de charge mentale subjective quel que soit niveau d'étude du sujet.
- H_1 Le niveau de charge mentale subjective est, en moyenne, significativement plus faible pour un niveau d'étude que pour l'autre.

Test d'hypothèse

Contrairement au premier facteur confondant, le niveau d'étude semble influencer la charge mentale subjective. En réalité, le résultat du test Mann-Whitney est égale à 0.09751. Nous remarquons

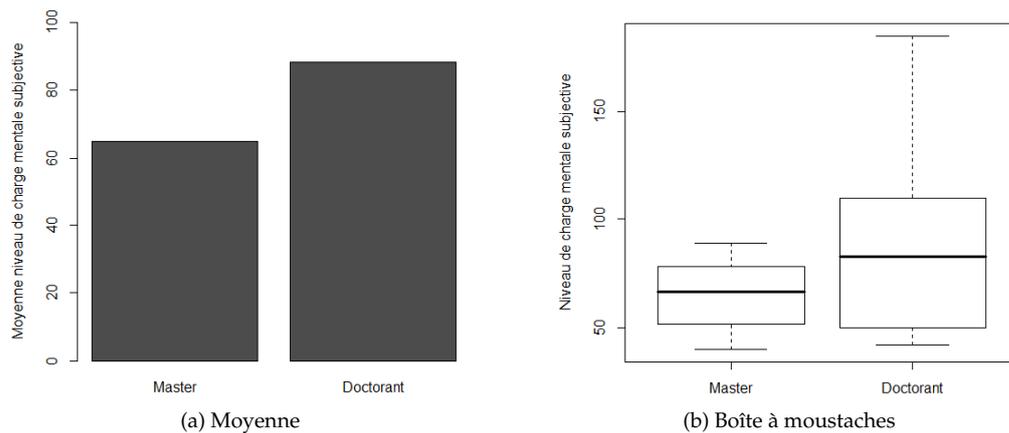


FIGURE 13.4 – Étude de la charge mentale subjective en fonction du niveau d'étude des sujets

donc une influence légèrement significative, si l'on considère un $\alpha = 0.1$. Les moyennes, de la figure 13.4a, illustrent une charge mentale plus faible chez les étudiants de master. Ainsi, nous pouvons rejeter l'hypothèse H_0 et affirmer la conclusion suivante (avec une probabilité d'erreur inférieure à 10%) :

Conclusion 44. *Le niveau de charge mentale subjective est, en moyenne, significativement plus faible pour les étudiants de master que pour les étudiants de doctorat.*

13.4 Conclusions

Ce chapitre contient l'ensemble des analyses portant sur la métrique d'évaluation de la charge mentale subjective. L'étude de l'influence des variables indépendantes (framework et représentation) ne permet pas d'établir de conclusion. Par ailleurs, l'analyse des facteurs confondants contient un résultat significatif. Ainsi, la seule conclusion propre à la charge mentale subjective se rapporte à l'influence du niveau d'étude. En d'autres termes, la conclusion 44 démontre une charge mentale significativement plus faible chez les étudiants de master, lorsqu'ils réalisent les tâches de composition de patrons, de participation des classes et d'identification des rôles.

Post expérience

Une fois les analyses produites pour chaque partie de l'expérience, il est important de réaliser une activité de rétrospection sur celle-ci. Comme son nom l'indique, ce chapitre passe en revue différents points inhérents à la partie post-expérience de notre expérience.

La section 14.1 présentent une comparaison des résultats suivant les différentes tâches. Ensuite, une comparaison des conclusions de notre expérimentation avec celles de Cepeda Porras est réalisée en section 14.2. Les différents résultats de la nouvelle représentation, nommée SPE, sont résumées en section 14.3. Par ce procédé, nous réalisons une évaluation de celle-ci. Enfin, les différentes menaces à la validité de notre expérience, et donc à la validité de nos conclusions, sont énoncées au sein de la section 14.4.

Sommaire

14.1	Comparaison des résultats des tâches	160
14.1.1	Résultats de la tâche « composition »	160
14.1.2	Résultats de la tâche « participation »	160
14.1.3	Résultats de la tâche « rôle »	161
14.1.4	Résultats de la mesure subjective	161
14.1.5	Résultats de la métrique de distance relative entre chemins visuels	161
14.2	Comparaison avec l'expérience de Porras	161
14.2.1	Tâche « composition »	162
14.2.2	Tâche « participation »	162
14.2.3	Tâche « rôle »	162
14.3	Évaluation de la représentation proposée	162
14.3.1	Tâche « composition »	163
14.3.2	Tâche « participation »	163
14.3.3	Tâche « rôle »	163
14.4	Menaces à la validité	164
14.4.1	Validité interne	164
14.4.2	Validité externe	166
14.4.3	Validité de construction	166
14.4.4	Validité de conclusion	167

14.1 Comparaison des résultats des tâches

Les chapitres 10, 11 et 12 contiennent les différentes parties de l'analyse statistiques des résultats de l'expérience. Ces différentes analyses nous ont permis d'établir plusieurs conclusions. Cette section a pour but de résumer ces conclusions. Afin de suivre la structure logique des chapitres 10, 11 et 12 qui divisent les analyses suivant la tâche donnée, nous subdivisons cette section de la même manière. Une sous-section est également consacrée à la synthèse des résultats de la métrique de distance relative entre chemins visuels.

14.1.1 Résultats de la tâche « composition »

L'analyse de la première tâche (« composition ») détermine plusieurs conclusions d'influence et de non-influence. Tout d'abord, les résultats démontrent que le framework ne possède aucun impact sur l'exactitude des réponses et sur le nombre moyen de fixation post-AORI, lorsque le sujet recherche les informations de réponse. Aucune influence du framework n'apparaît sur la durée moyenne des fixations lorsqu'un sujet rassemble les éléments de réponse de la tâche considérée.

En ce qui concerne les résultats relatifs à la représentation utilisée, ceux-ci démontrent par plusieurs métriques qu'il existe des effets significatifs dus à la nature de la représentation.

Dong semble être la représentation la plus appropriée pour rechercher les patrons dans lesquels une classe déterminée intervient. En effet, dès que les sujets ont repéré la classe concernée, il ne leur reste plus qu'à lire le stéréotype pour trouver la réponse. Cette représentation nécessite plus de traitement qu'UMLNotation, mais moins de recherche. Ainsi, avec Dong les sujets regardent plus souvent dans les zones pertinentes qu'avec SPE. De plus, les sujets utilisant Dong démontrent une meilleure interprétation des zones d'intérêt que ceux utilisant UMLNotation ou SPE. Cette observation s'explique par un mélange des informations des patrons et des informations du diagramme de classes chez UMLNotation et une difficulté à différencier les couleurs chez SPE.

Enfin, lorsque les sujets recherchent les patrons dans lesquels intervient une classe, les étudiants de masters se comportent de façon plus homogène que les doctorants.

14.1.2 Résultats de la tâche « participation »

Les données obtenues au cours du deuxième type de tâche relèvent une influence du framework sur le temps nécessaire à la recherche des éléments de réponse et une légère influence sur le taux IN AORI/IN AOII normalisé. La dernière conclusion sur le framework démontre une influence sur la durée moyenne des fixations lorsque que le sujet rassemble les éléments de réponse.

Pour ce type de tâche, les sujets démontrent un effort moyen moindre lorsqu'ils utilisent UMLNotation. En réalité, les sujets, utilisant UMLNotation, procèdent à des recherches directes et concentrées sur des zones pertinentes. D'après cette conclusion, il est probable que le ratio durée de fixation/saccade significativement plus faible pour UMLNotation soit dû à un temps de traitement plus faible. De plus, la partie « support » de la tâche révèle un taux IN AORI/IN AOII normalisé supérieur pour UMLNotation. Lorsqu'ils sont comparés à UMLNotation, Dong et SPE semblent souffrir d'un manque de visibilité ou d'un manque de mise en évidence des zones pertinentes. Pour SPE, ce phénomène est caractérisé par le problème de différenciation des couleurs. En outre, les sujets ayant utilisé SPE dénotent une recherche directe et une concentration plus régulière sur les zones d'intérêt que les participants ayant utilisé Dong. La notation Dong est donc la représentation la moins intéressante lorsqu'il s'agit d'identifier toutes les classes participant à un patron de conception.

Pour finir, la moyenne des distances relatives entre chemins visuels pour le niveau d'étude des

participants révèle la même conclusion que pour la tâche « composition ». En effet, il s'avère que les étudiants de master ont un parcours visuel plus homogène que les doctorants.

14.1.3 Résultats de la tâche « rôle »

La tâche d'identification des rôles d'une classe démontre également plusieurs conclusions. Premièrement, le framework influence la densité de transition, mais n'influence pas le taux de validité des réponses.

Deuxièmement, la représentation n'influence pas, pour la partie question, la densité de transition et le taux IN AORI/IN AOII normalisé.

Troisièmement, UMLNotation semble être la représentation la moins intéressante en cas d'identification des rôles d'une classe. En effet, les sujets ayant utilisé cette représentation exposent une durée des fixations faible, un taux de transition AORI vers AOII élevé et un ratio durée de fixation/saccade faible. Ces informations nous laissent penser qu'avec UMLNotation, les informations des rôles sont lisibles, mais difficilement distinguables des rôles attachés aux associations des diagrammes de classes. C'est pourquoi, le sujet doit opérer beaucoup plus de recherches et doit comparer les zones non pertinentes avec les zones pertinentes pour trouver la réponse. Ces difficultés sont illustrées par un taux de bonnes réponses plus faible chez les sujets utilisant UMLNotation que chez les sujets utilisant SPE.

Enfin, les résultats démontrent que les femmes ont, en moyenne, une distance relative entre chemins visuels significativement plus faible que les hommes.

14.1.4 Résultats de la mesure subjective

La dernière partie de l'analyse des résultats concerne l'évaluation subjective de la charge mentale. Aucune conclusion ne peut être tirée concernant l'influence ou la non influence des variables indépendantes. Par ailleurs, l'analyse des facteurs confondants contient un résultat significatif sur l'influence du niveau d'étude des participants. D'après les analyses, les masters présentent une charge mentale significativement plus faible que les doctorants lorsqu'ils réalisent des tâches de composition de patrons, de participation de classes et d'identification des rôles.

14.1.5 Résultats de la métrique de distance relative entre chemins visuels

Les résultats trouvés pour la métrique de distance relative entre chemins visuels exposent le fait que les sujets se comportent de façon différente en fonction de leur expérience. Cette découverte avait déjà été relevée par De Smet et Lempereur [DL11] et conforte, une fois de plus, l'observation faite par Yusuf *et al.* [YKM07]. En réalité, les doctorants utilisent leurs expériences personnelles pour parcourir le diagramme de classes et démontrent donc moins d'homogénéité que les masters qui ont majoritairement une expérience académique similaire.

14.2 Comparaison avec l'expérience de Porras

Comme nous l'avons déjà annoncé en section 9.1, notre expérience se base sur une expérimentation précédemment réalisée par Cepeda Porras [Por08]. La conception que nous avons choisie n'est pas exactement la même que celle de Porras. Néanmoins, nous pouvons établir une comparaison des

conclusions des deux expériences. Cette section met en place une comparaison des représentations UMLNotation et Dong, pour chaque tâche.

14.2.1 Tâche « composition »

Les deux expériences illustrent bien une différence significative entre UMLNotation et Dong au niveau de la pertinence de l'effort et de la difficulté à trouver les classes essentielles. Pour ces métriques, les sujets sont les plus efficaces lorsqu'ils utilisent Dong. Par ailleurs, notre expérience permet de relever une meilleure interprétation des zones d'intérêt lorsque les sujets utilisent Dong. Un ratio durée fixation/saccade significativement plus faible a également été remarqué chez les sujets utilisant UMLNotation. Ce phénomène s'explique par une recherche concentrée sur les noms des classes et donc moins de traitement nécessaire à la lecture du stéréotype de Dong. Enfin, Cepeda Porras relevait également un meilleur pourcentage de réponses correctes avec Dong ; toutefois cette observation n'est pas présente au sein de notre expérience.

14.2.2 Tâche « participation »

Comme relevé dans l'expérience de Porras, il s'avère qu'UMLNotation se distingue du point de vue de la performance et de la régularité de passage dans les zones pertinentes. Avec UMLNotation, les sujets ont besoin de moins de temps, ont un taux de réponses exactes plus élevé et regardent plus souvent dans des zones d'intérêts pertinentes.

Notre expérience permet de relever le fait que les sujets utilisant UMLNotation :

- Nécessitent une étendue de recherche significativement plus faible que ceux utilisant Dong.
- Ont une meilleure visibilité des zones d'intérêt pertinentes que ceux utilisant Dong.
- Nécessitent moins de temps de traitement que ceux utilisant Dong.
- Se concentrent plus souvent sur des zones d'intérêt pertinentes que ceux utilisant Dong, lorsqu'ils recherchent et qu'ils rassemblent les éléments de réponses.

14.2.3 Tâche « rôle »

Contrairement à l'expérience précédente, nous ne pouvons pas établir de conclusions concernant les métriques de performance pour la comparaison de Dong et UMLNotation. Cependant, cette année, UMLNotation présente un temps moyen de fixation significativement plus petit que Dong. Le ratio de la durée fixation/saccade inférieur chez UMLNotation montre qu'avec cette représentation, l'identification des rôles d'une classe nécessite plus de recherche que pour Dong. Par contre, avec UMLNotation, les sujets possèdent un pourcentage de transition des zones d'intérêt vers les zones de non-intérêt le plus élevé. Par conséquent, nous pouvons émettre l'hypothèse qu'UMLNotation souffre d'une mauvaise interprétation de ses zones pertinentes.

14.3 Évaluation de la représentation proposée

La section 3.2 présente une nouvelle représentation de patron de conception, sous le nom de stéréotype *pattern enhanced notation*. L'expérience mise en place cette année nous permet de mettre en avant les avantages et les inconvénients de cette nouvelle représentation. Les différentes remarques sont classifiées, ci-dessous, selon la tâche qui est demandée au sujet.

14.3.1 Tâche « composition »

Pour cette tâche, les sujets ayant utilisé SPE démontrent une pertinence de l'effort plus petite que ceux ayant utilisé Dong. La présence de couleurs difficiles à différencier est la cause directe de cette différence significative vis-à-vis de Dong. Ce problème de couleur est également remarquable à partir de la métrique du pourcentage de transition des zones pertinentes vers les zones non pertinentes. Pour cette métrique, SPE possède une valeur significativement plus élevée que Dong. Par conséquent, cette difficulté diminue les qualités de SPE qui normalement devrait permettre au sujet d'avoir une meilleure performance que Dong, pour la tâche de composition de patrons.

14.3.2 Tâche « participation »

C'est au sein de cette tâche que la nouvelle représentation proposée était censée offrir ses avantages. Malheureusement, le problème des différentes couleurs a reproduit une augmentation de l'effort cognitif des sujets. Cette observation est à nouveau présente dans les résultats de certaines métriques.

Premièrement, SPE possède une étendue de recherche significativement inférieure à UML-Notation. Deuxièmement, SPE possède une densité de transition significativement supérieure à UMLNotation, mais significativement inférieure à Dong. Troisièmement, pour la partie « question », SPE possède un taux IN AORI/IN AOII normalisé plus faible qu'UMLNotation, mais plus élevé que Dong. Pour la partie support, les sujets utilisant SPE sont passés moins souvent dans les zones pertinentes que les sujets utilisant UMLNotation. De plus, SPE possède un pourcentage de transition AORI vers AOII significativement plus élevé que Dong et UMLNotation. Ces résultats montrent bien que la différenciation des couleurs chez SPE nécessite de reproduire plusieurs fois des saccades identiques vers des zones sans intérêt. Enfin, lorsque l'on compare SPE et UMLNotation, la première représentation possède une valeur significativement plus élevée pour le nombre moyen de fixations post-AORI.

Ces résultats d'analyses montrent bien que le problème des couleurs a transformé les avantages de SPE en inconvénients et qu'il est crucial de penser à sélectionner les couleurs suivant une métrique bien précise. Il est également important de soulever le fait qu'un trait plus épais pour les couleurs faciliterait la différenciation dans le diagramme de classes.

14.3.3 Tâche « rôle »

Dong et SPE possèdent des stéréotypes contenant les mêmes informations textuelles. Une réduction de ce texte a été opérée dans la nouvelle représentation, afin de réduire l'effort cognitif des sujets. Malheureusement, aucune différence significative n'existe, entre SPE et Dong, pour exprimer cet avantage potentiel. Nous ne pouvons donc établir aucune conclusion concernant les bénéfices de cet ajout.

Cependant, nous pouvons quand même relever une différence significative pour le taux de réponses exactes. En effet, les sujets utilisant SPE possèdent un taux de réponses exactes significativement plus élevé que les participants utilisant UMLNotation. Tout comme Dong, SPE est donc une représentation intéressante à utiliser lorsque l'on cherche à identifier les rôles joués par une classe.

14.4 Menaces à la validité

Cette section a pour but de présenter les différentes menaces à la validité que notre expérience possède, et dans quelle mesure nous les acceptons. Quelques menaces sont liées à l'utilisation de sujets humains dans l'étude, d'autres à l'utilisation de l'équipement oculométrique. Comme expliqué dans la section 6.3.3, les menaces à la validité couvrent quatre parties :

- **La validité interne** : La causalité de la relation entre le traitement et les résultats fait partie de la validité interne. Seul le traitement doit avoir produit le résultat.
- **La validité externe** : La généralisation des résultats concerne ce type de menace. Les résultats peuvent-ils être généralisés en dehors du cadre de celle-ci s'il existe bien une relation causale entre la construction de la cause et l'effet ?
- **La validité de construction** : Cette validité concerne la relation entre la théorie et l'observation. Si la relation entre l'effet et la cause est bien causale, il faut s'assurer que le traitement reflète correctement la construction de la cause et que le résultat reflète correctement la construction de l'effet.
- **La validité de conclusion** : Elle est concernée par la relation entre le traitement et les résultats. Il faut s'assurer qu'il y ait une relation statistique significative.

Il est évident que la structure de cette section s'inspire de ces quatre validités. Ainsi, la validité interne sera évoquée en premier, suivie de la validité externe, la validité de construction prendra ensuite place et enfin nous terminerons cette section en présentant la validité de conclusion.

14.4.1 Validité interne

Parmi les menaces à la validité interne, nous avons retenu quatre menaces qui peuvent potentiellement être présentes dans notre expérimentation. Les menaces de la maturation (représentée par la courbe d'apprentissage et la fatigue) et l'instrumentation ont été acceptées après une réduction. La menace de diffusion des traitements a elle, été écartée.

La maturation

La première menace est la menace de *maturation*. Cette menace représente le fait que le sujet est affecté négativement (fatigue) ou positivement (effet d'apprentissage) au fur et à mesure de l'expérience.

La première partie de cette menace est donc *la courbe d'apprentissage*. Notre expérience peut être affectée par cette menace, car elle demande des tâches assez répétitives aux sujets. Ceci a pour conséquence de modifier la courbe d'apprentissage du sujet. L'expérience est susceptible à cette menace car elle demande des tâches assez répétitives aux sujets. Or, cette courbe influence directement les résultats des sujets aux différentes questions. Par cette influence, et par le fait que les différentes analyses utilisent une métrique qui reprend le taux de réponses correct, cette courbe d'apprentissage touche également aux résultats même de l'expérimentation. Concrètement, cet apprentissage peut apparaître au fur et à mesure de la réalisation des tâches sur un même diagramme. Le fait d'utiliser deux frameworks augmente également les chances d'apparition de cet apprentissage.

Du fait que l'effet d'apprentissage est fortement réduit lorsque le sujet passe au deuxième framework de l'expérience, le choix de faire un roulement dans l'ordre des frameworks s'est imposé. Cependant, l'ordre des tâches est resté le même pour tous les sujets. En fait, l'ordre choisi n'est pas

anodin. Le questionnaire commence avec des questions assez générales (tâches « composition » et « participation »). Les questions portant sur la tâche « rôle », et donc, plus précises sont placées en dernier lieu. Via ce mécanisme, la courbe d'apprentissage est contrôlée autant que possible afin de la rendre équivalente pour tous les sujets.

Enfin, le choix des projets logiciels et les types de tâches retenus peuvent également influencer la courbe d'apprentissage des sujets. Afin de réduire au maximum cette menace, nous avons pris soin de ne pas choisir de projets logiciels étudiés par le laboratoire PtiDej ou utilisés dans une précédente expérimentation. En effet, comme montré dans la section 9.1.1, Nutch et JRefactory ont été choisis en respectant un certain nombre de contraintes.

De plus, les types des tâches des différentes questions étaient inédites dans l'École Polytechnique de Montréal.

La deuxième partie de cette menace est représentée par la *fatigue*. Comme souvent dans les expériences qui utilisent un eye-tracker, la fatigue ressentie par un sujet durant l'expérience et la diminution de la concentration représente une menace à la validité interne. Les questions demandées aux sujets étaient des questions relativement simples et courtes¹. Dans le but de limiter les effets de la fatigue, une pause était faite entre les deux frameworks afin que le sujet ait la possibilité de reposer ses yeux ou de boire. Cependant, peu de sujets ont vraiment pris la peine de faire une réelle pause. Les raisons évoquées par les intéressés étaient qu'ils ne se sentaient pas fatigués.

Grâce à ces précautions, la durée de passage d'un sujet variait entre 25 et 40 minutes. Aucun sujet n'a dépassé une heure d'expérimentation.

L'instrumentation

Le fait d'utiliser un oculomètre implique également des risques vis-à-vis des résultats obtenus. Malgré les phases de calibrations et de tests, des offsets apparaissaient régulièrement durant les enregistrements. Ces offsets proviennent de mouvements de la tête produits par le sujet ou proviennent d'une mauvaise détection par l'oculomètre. Les mouvements de la tête sont parfois inévitables. En effet, lorsque les sujets se concentrent pour donner la réponse, ils oublient souvent la restriction du mouvement de la tête. Pour minimiser cette menace, le sujet est placé sur une chaise de dentiste. Celle-ci favorise une position fixe, diminuant ainsi des mouvements du corps qui sont inappropriés aux oculomètres. De plus, des images contenant des points à fixer ont été utilisées afin de détecter plus facilement les offsets. Les données oculométriques obtenues sur ces images nous ont permis d'optimiser la correction effectuée via l'outil mis en place dans TAUPE.

Nous avons aussi identifié et accepté une menace à l'instrumentation causée par la résolution maximale possible pour l'affichage des diagrammes (1920 * 1080 pixels) dans notre système. Même s'il s'agit d'une bonne résolution, elle empêche la création de plus grand diagramme.

La diffusion des traitements

Finalement, pour éviter la diffusion des informations de l'expérience et donc empêcher des sujets d'apprendre les traitements d'avance, nous avons demandé à chacun de ne pas parler de l'expérience avant la date de la fin de celles-ci. Nous croyons que les sujets ont suivi ces instructions.

1. D'ailleurs, une analyse de l'expérience réalisée par De Smet et Lempereur [DL11] a montré que, en plus d'être plus fatigantes pour les sujets, les questions complexes n'apportaient pas plus d'informations pertinentes que les questions simples.

14.4.2 Validité externe

Deux menaces limitent la possibilité de généraliser nos résultats pour une pratique industrielle : l'interaction de sélection et l'interaction du réglage. Malgré le fait qu'elles ont été amoindries, ces deux menaces nécessitent d'être acceptées.

L'interaction de sélection

Un manque d'hétérogénéité des sujets participant à l'expérience se traduit par une menace que l'on nomme *l'interaction de sélection*. En effet, les sujets sont tous des étudiants de deuxième cycle ou troisième cycle. De plus, ils ont tous une connaissance suffisante du langage UML et des patrons de conception. Ils ne sont donc pas représentatifs de l'industrie du logiciel. Cependant, bien que l'homogénéité des sujets diminue la validité externe, celle-ci augmente la validité de conclusion.

L'interaction du réglage

La menace d'*interaction du réglage* est en relation avec la taille et la complexité des diagrammes utilisés. Afin d'affaiblir cette menace, nous avons pris deux frameworks (JRefactory et Nutch) qui utilisent amplement les patrons de conception. Les deux frameworks contiennent le même nombre de patrons de conception. De plus, nous avons limité le nombre de classes à 20 pour ne pas rendre la complexité trop grande. Cette menace était déjà présente dans l'expérimentation de Cepeda Porras [Por08, p.57]. Son expérience était réalisée avec le framework JHotDraw possédant les mêmes caractéristiques que ceux utilisés ici. Cette menace avait induit la remarque suivante :

« Cependant, nous ne pouvons pas être certains que nos résultats peuvent s'appliquer à tous les systèmes, des répliques devront être menées pour aboutir à des conclusions plus générales. »

[Por08]

Vu que nous arrivons aux mêmes conclusions avec deux autres projets logiciels (cf. section 14.2), cette menace a été amoindrie. Néanmoins, la généralisation de nos résultats nécessiterait de reproduire l'expérience sur des frameworks industriels réels. Malheureusement, ce type d'expérience est souvent irréalisable et extrêmement coûteuse.

14.4.3 Validité de construction

Nous avons identifié et traité quatre menaces à la validité de construction : le biais mono-méthode, le biais mono-opération, l'estimation d'hypothèse et l'appréhension. Par ailleurs, nous avons identifié et accepté, tout en la diminuant, la menace attentes des expérimentateurs.

Les attentes des expérimentateurs

En tant qu'expérimentateurs, nous avons conçu les diagrammes et les questions. Afin de créer ces questions, nous avons dû prendre connaissance des représentations ainsi que des différentes expériences réalisées sur celles-ci. Or, par ce procédé nous avons aussi pris connaissance des forces et des faiblesses de ces différentes représentations. À partir de cette connaissance, il est probable que nous ayons introduit un biais involontaire dans les diagrammes et les questions. En effet, il est tout à fait possible que nous ayons créé inconsciemment des questions et des tâches plus faciles à réaliser avec une certaine représentation. Toutefois, les questions ont préalablement été révisées et

supervisées par des personnes ne participant pas à l'expérience, afin de diminuer notre influence sur celles-ci.

Pour réduire d'avantage cette menace, les diagrammes ont été initialement obtenus par une méthode de rétro-ingénierie telle que décrite en section 9.1.1.

Le biais mono-méthode

Notre expérimentation écarte cette menace par le fait qu'elle emploie une multitude de métriques. En effet, jusqu'à neuf variables dépendantes sont utilisées lors des différentes analyses (cf. section 9.1.8).

Le biais mono-opération

Cette menace a également été traitée par le fait que notre expérience ne se contente pas d'un seul framework, mais de deux.

L'estimation d'hypothèses

Pour la menace *d'estimation des hypothèses*, nous avons procédé comme pour l'expérience de Cepeda Porras [Por08]. Concrètement, nous avons décidé de ne pas informer les sujets du but de l'expérience. Cette information n'a été divulguée que lors de l'exécution de celle-ci par le sujet. En effet, c'est durant la lecture des différents tutoriels qu'ils ont pris connaissance des tâches qu'ils allaient effectuer.

L'appréhension

L'appréhension des sujets est une menace difficile à maîtriser du fait qu'elle est indépendante de l'expérience.

Premièrement, afin d'atténuer l'appréhension des sujets n'ayant jamais eu affaire avec un oculomètre, nous les avons rassurés en promettant que le matériel était indolore et non-intrusif².

Deuxièmement, certains sujets avaient une crainte vis-à-vis de leur connaissance pour répondre aux questions de l'expérience, et ce, a posteriori de celle-ci. La garantie de garder les données anonymes a permis de rassurer un bon nombre d'entre eux. De plus, au vu de la *simplicité* des questions, l'ensemble des sujets ont eu l'air satisfaits de leur expérience³.

Troisièmement, pour éviter un effet malsain d'oppression, aucune limite temporelle n'a été instaurée. Même si le temps utilisé par les sujets fait partie d'un élément participant aux analyses de performance, nous avons simplement demandé au sujet de prendre le temps qui leur semblait nécessaire pour répondre à chaque question.

Dernièrement, les sujets étaient principalement des personnes du laboratoire PtiDej, ce qui implique que la plupart de ceux-ci n'étaient ni impressionnés, ni intimidés par une telle expérience.

14.4.4 Validité de conclusion

Pour la validité de conclusion, trois menaces sont présentes : la fiabilité des mesures, les insignifiances aléatoires des réglages de l'expérience et l'hétérogénéité aléatoire des sujets. Seul la menace de fiabilité des mesures n'a pas pu être totalement éliminée.

2. Un sujet avait eu une mauvaise expérience avec l'oculomètre EyeLinkII et s'est vu rassuré quand nous lui avons présenté FaceLAB.

3. La difficulté de l'expérience de De Smet et Lempereur [DL11] semble avoir eu un effet néfaste quant à l'estimation de leur connaissance.

La fiabilité des mesures

Comme nous l'avons expliqué dans la section, l'utilisation d'un oculomètre entraîne l'apparition de certains offsets. Le problème est qu'il est difficile, voire impossible de les éviter et de les prévoir. De plus, si certains offsets sont faciles à corriger, d'autres sont plus compliquées à repositionner. Dans ces cas là, il devient difficile d'apporter une correction *sûre*, c'est-à-dire de savoir réellement où la fixation devrait se situer. C'est ce problème qui nous a obligé à retirer quelques sujets. En effet, cette menace était trop importante sur certains sujets. Seuls persistent les sujets qui ont eu de légers offsets. Cependant, vu qu'une correction manuelle a dû avoir lieu, nous ne pouvons pas rejeter complètement cette menace.

Les clignements des yeux des sujets peuvent également influencer les fixations de manière anecdotique. Les fixations manquantes durant les clignements des yeux d'un sujet ne sont pas à prendre en compte étant donné que le sujet ne visualise pas le diagramme durant ces microsecondes. Cette source est donc écartée.

Le choix des termes utilisés pour formuler les tâches à effectuer peut être vu comme une menace à la fiabilité des mesures. En effet, si le sujet ne comprend pas correctement la question, des biais peuvent apparaître. Enfin, le questionnaire était disponible en plusieurs langues (anglais/français) et nous étions présents durant toute la durée de l'expérience afin de répondre aux éventuelles questions des sujets. Par ces différents mécanismes, nous avons réduit l'importance de cette menace.

Les insignifiances aléatoires des réglages de l'expérience

Pour traiter les insignifiances aléatoires des réglages de l'expérience, nous avons utilisé un laboratoire calme pour réaliser les expériences sans aucune distraction. Nous avons aussi fait des tests préliminaires d'exécution avant de commencer les expériences de façon officielle.

L'hétérogénéité aléatoire des sujets

L'hétérogénéité des sujets est un facteur présent dans toute expérience. Le principal problème de cette menace est le fait qu'un groupe fortement hétérogène peut biaiser les observations, et donc les résultats. En effet, on peut se retrouver dans une situation où les observations sont plus influencées par les différences des sujets que pas les représentations (la variable indépendante étudiée). Dans notre cas, nous nous sommes assurés d'avoir un ensemble de sujets relativement uniforme, ce qui écarte cette menace.

Troisième partie

Conclusion

Conclusions et travaux futurs

15.1 Réponses aux questions d'exploration, de recherche et de prospection

Cette section a pour but de rassembler les informations pertinentes que l'*Analyse de la compréhension des design patterns : expérimentation avec différentes représentations* a fait ressortir.

Ces informations pertinentes proviennent des réflexions effectuées afin de répondre aux différentes questions, précédemment présentées au sein du chapitre d'introduction. Leurs réponses ont été décrites tout au long de ce mémoire. Par conséquent, celles-ci ne sont résumées qu'au sein de cette section, avec un rappel préalable de la question concernée.

15.1.1 Questions d'exploration

Avant de pouvoir débiter notre expérimentation, il a été nécessaire d'effectuer un travail de recherche et de synthèse afin d'établir les bases de celle-ci. Ces travaux nous ont permis de répondre aux différentes questions d'exploration.

Question 1 : Quelles théories et outils le génie logiciel offre-t-il pour mesurer la compréhension de la structure d'un programme contenant des design patterns ?

Les éléments de la réponse à cette question trouvent leur source dans le chapitre qui est entièrement dédié à la compréhension, c'est-à-dire le chapitre 3. En effet, avant de pouvoir répondre d'une manière concrète à cette question, il est important de bien définir le terme « compréhension ».

Une fois ce terme défini, ce chapitre se consacre à la présentation des mécanismes aidant le programmeur dans la tâche de compréhension de programme. Par ailleurs, ces programmes utilisent couramment des patrons de conception. Bien qu'il existe une multitude de modèle ([Mar95] ou [Rug95]) permettant d'identifier et d'optimiser la compréhension des développeurs dans un projet informatique, les modèles mettant en avant la compréhension des structures contenant des design patterns sont beaucoup plus rares. Or, comme nous l'avons expliqué dans le chapitre 2, les patrons de conception possèdent des propriétés favorables à la compréhension de programmes, comme par exemple : un vocabulaire commun, une documentation du logiciel et une structure propice au changement.

Plusieurs méthodes peuvent être employées pour évaluer l'effort ou la charge mentale nécessaire au sujet pour réaliser les tâches de compréhension. Parmi celles-ci, nous avons décidé d'en choisir trois : l'analyse de la performance, l'analyse des mesures physiologiques et l'analyse d'une mesure

subjective. Alors que la première méthode est couramment employée, les deux autres sont moins ordinaires. L'utilisation de mesures physiologiques nécessitent la connaissance de théories auxiliaires, comme celle de la science de la vision, et la connaissance de certaines métriques. Ces éléments sont présentés et utilisés au cours des différentes analyses. La mesure subjective se base sur NASA-TLX, qui est un système subjectif d'évaluation de la charge mentale.

Question 2 : Comment réaliser une approche empirique pour évaluer la compréhension ?

Pour la réalisation de l'étude empirique, nous nous sommes inspirés de la théorie expérimentale de Wholin *et al.* [WRH+00]. En effet, ils proposent une approche connue et reconnue dans le domaine du génie logiciel. Le chapitre 6 reprend cette théorie et la résume.

Vu que nous suivons cette démarche, notre expérimentation a suivi le processus *conseillé* :

1. *Définition de l'expérience.*
2. *Planification de l'expérience.*
3. *Exécution de l'expérience.*
4. *Analyse et interprétation des données.*

La définition de l'expérience, la planification et l'exécution de celle-ci suivent de près le procédé classique des expériences du génie logiciel. Ces différentes étapes sont d'ailleurs présentées et détaillées dans le chapitre 9.

L'objectif principal de notre expérimentation est d'évaluer la charge mentale des sujets lorsque ceux-ci réalisent des tâches qui demandent une compréhension de la structure du programme. L'évaluation de cette charge mentale est réalisé par plusieurs méthodes. La première, nommée *mesure de la performance*, se concentre sur le pourcentage des réponses correctes ainsi que le temps pris pour répondre. Il s'agit donc de savoir d'une manière générale si le sujet répond correctement ou non à cette tâche. De plus, le temps peut donner une information pertinente sur la réflexion du sujet. Ensuite, arrive la *mesure des données physiologiques*. Cette mesure s'intéresse plus particulièrement au parcours visuel du sujet. Il s'agit en fait d'exploiter d'une manière pratique la théorie de la Vision-Compréhension (présenté par Guéhéneuc [Gué08]). Un total de huit métriques, définies dans le chapitre 4 et dont chaque signification est expliquée dans le chapitre 9, sont utilisées. En dernier lieu arrive la *mesure subjective* que nous effectuons avec une méthode largement utilisée, nommée NASA-TLX (dont la définition se trouve au chapitre 5).

Question 3 : Quelles alternatives et variantes existe-t-il à la représentation des design patterns dans des diagrammes de classe UML ?

Afin de mesurer la compréhension de la structure d'un programme contenant des designs patterns, il faut avoir des supports qui représentent ces structures. Actuellement, la représentation standard de visualisation de patrons de conception est UML collaboration notation.

Comparé à cette représentation, il existe quelques autres supports, dont on suppose que la nature même du diagramme fait varier la charge mentale demandée pour la compréhension. Ces représentations sont détaillées dans le chapitre 3 et ne sont donc pas ré-expliquées au sein de cette section.

Le nombre de ces représentations est assez restreintes. De plus, celles-ci souffrent également d'un manque d'étude les concernant. Cependant, une étude de différentes représentations réalisée par

Cepeda Porras convenait pour notre sujet. En effet, le but de son expérience a été de vérifier s'il existe effectivement une diminution de la charge cognitive et donc une amélioration à la compréhension de programmes pendant l'exécution de tâches basiques à la compréhension de patrons de conception [Por08]. Ci-dessous, les représentations retenues pour son expérience :

- La représentation standard, nommée « UML Collaboration Notation » [Vli98].
- La représentation « stereotype-enhanced UML diagrams » de Dong *et al.* [DYZ07].
- La représentation « Pattern-enhanced class diagrams » de Schauer et Keller [SK98].
- La représentation « pattern :role » notation de Gamma *et al.* [Gam96].

Certaines représentations, présentées dans ce document, ne font pas parties de notre expérimentation (la représentation « Pattern-enhanced class diagrams » de Schauer et Keller et la représentation « pattern :role » notation de Gamma *et al.*). Ce choix est motivé par les faibles résultats obtenus avec ces présentations dans l'étude de Cepeda Porras [Por08]. Enfin, une nouvelle représentation reprenant l'aspect visuel de Schauer et Keller [SK98] et l'aspect textuel de Dong *et al.* [DYZ07], a été créée par nos soins (nous l'avons nommée « Stereotype Pattern Enhanced »). Ainsi, nous espérons assembler les avantages de ces aspects en supprimant leurs inconvénients.

Toutes ces représentations utilisent le langage UML. Ce caractère les rends particulièrement intéressantes pour une expérience car la majorité des personnes travaillant dans le génie logiciel connaissent ce langage.

15.1.2 Questions de recherche

Comme l'indique l'intitulé de ce document, « *Analyse de la compréhension des patrons de conception : expérimentation avec différentes représentations* », nos questions de recherche se sont articulées autour du lien entre une représentation particulière et l'efficacité de réalisation d'une tâche de compréhension. Nous avons pris soin de réaliser une étude qui puisse apporter des explications plus détaillées quant à l'expérimentation de Cepeda Porras tout en offrant la possibilité qu'elle puisse offrir des conclusions inédites.

Question 4 : Est-ce que notre étude confirme l'influence des représentations détectée par Cepeda Porras ?

D'après l'étude réalisé par Cepeda Porras [Por08], ces représentations ont un impact sur la compréhension des programmes. Il était arrivé à la conclusion suivante :

« Pour conclure, Dong avec sa représentation fortement textuelle, mais aussi avec sa richesse sémantique est plus efficace pour les tâches de composition de patrons et d'identification de rôles (tâche « rôle ») que la représentation de diagrammes de collaboration UML quand le nombre de classes du diagramme n'est pas trop grand. Il faut aussi remarquer que la représentation de Dong ajoute d'autres informations que les autres représentations.

La représentation UML collaboration notation (et la représentation de diagrammes de classes augmenté de patrons de Schauer et Keller), en raison de son contenu plus graphique sont plus efficaces pour repérer les classes participant (tâche « participation ») dans un patron de conception indépendamment de la densité de classes. »

[Por08]

Notre étude reprend ces deux représentations. Plusieurs résultats significatifs ont été trouvés. Vu que notre expérimentation a, comme pour celle de Porras, divisé les résultats par type de tâche, nous les présentons ici également suivant cette structure.

Pour la tâche « composition », le sujet a plus facile de la réaliser avec la représentation de Dong qu'avec UML Collaboration Notation.

Pour la tâche « participation », la représentation UML Collaboration Notation est plus utile que la représentation de Dong.

Enfin, en ce qui concerne la tâche « rôle », il est probable que la représentation UML Collaboration Notation soit la moins intéressante.

Ces comparaisons sont détaillées dans le chapitre 14. Le lecteur peut remarquer que l'expérimentation confirme la conclusion émise par Cepeda Porras. Cette expérience a été réalisée avec des frameworks, des sujets et des questions différentes de Cepeda Porras.

La nouvelle représentation, basée sur l'utilisation des couleurs combinée aux textes offerts par Dong, n'a pas eu les résultats escomptés. En effet, celle-ci a des résultats significatifs montrant son inefficacité en terme de charge mentale pour les différentes tâches. La présence des couleurs sans la validation d'une quelconque métrique est une raison de cet échec.

Il n'existe donc aucune représentation idéale pour les trois tâches données. L'aspect visuel donne un avantage pour la tâche « participation » mais perd celui-ci au profit de l'aspect textuel pour la tâche « composition » et « rôle ».

Question 5 : Pour quelles raisons la représentation Dong est-elle moins performante que la représentation UML Collaboration Notation lorsqu'il s'agit d'identifier les classes appartenant à un design pattern alors qu'elle est plus performante pour détecter les design patterns dans lesquels une classe intervient ainsi que les différents rôles joués par une classe ?

Si notre étude confirme celle réalisée par Cepeda Porras [Por08], le but était surtout de comprendre pourquoi et donc d'apporter des explications. Un pré-requis afin de répondre à cette question était que notre étude arrive aux mêmes conclusions que celle de Porras. Pour apporter ces explications supplémentaires, nous avons utilisé une multitude de métriques lors de notre expérimentation.

Cette question peut être divisée en deux sous-questions. La première question est **Pour quelles raisons la représentation Dong est-elle moins performante que la représentation UML Collaboration Notation lorsqu'il s'agit d'identifier les classes appartenant à un design pattern (tâche « participation ») ?**

Pour la tâche « participation », UML Collaboration Notation se distingue du point de vue de la performance et de la régularité de passage dans les zones pertinentes. De plus, les sujets ont besoin de moins de temps, ont un taux de réponses exactes plus élevé et regardent plus souvent dans des zones d'intérêts pertinentes. En fait, notre étude a relevé que les sujets utilisant UML Collaboration Notation ont une densité spatiale et une densité de transition plus faibles. Ces conclusions permettent d'affirmer que les sujets, utilisant UML Collaboration Notation, procèdent à des recherches directes et concentrées sur des zones pertinentes. Comme la recherche est directe et concentrée avec UML Collaboration Notation, il est probable que cette différence soit due à un temps de traitement plus faible. Enfin, les sujets se concentrent plus souvent sur des zones d'intérêt pertinentes que

ceux utilisant Dong, lorsqu'ils recherchent et qu'ils rassemblent les éléments de réponses. Il est clair que Dong perd son efficacité par son surplus d'information où le nom du design pattern est *noyé*. Les informations étant contenue dans chaque classe, la recherche doit également être plus étendue. **Notre étude confirme donc la conclusion de Cepeda Porras pour cette tâche.**

La seconde question serait **Pour quelles raisons la représentation Dong est-elle plus performante que la représentation UML Collaboration Notation pour détecter les design patterns dans lesquels une classe intervient (tâche « composition ») ainsi que les différents rôles joués par une classe (tâche « rôle ») ?**

Dong semble être la représentation la plus appropriée pour la tâche « composition ». En effet, dès que les sujets ont repéré la classe concernée, il ne leur reste plus qu'à lire le stéréotype pour trouver la réponse. La métrique qui donne le taux de transition des zones d'intérêts vers les zones de non-intérêts montre que les utilisateurs d'UML Collaboration Notation ou de SPE ont un taux significativement plus élevé que les sujets utilisant Dong. Cette représentation nécessite plus de traitement qu'UML Collaboration Notation, mais moins de recherche. Ainsi, avec Dong les sujets regardent plus souvent dans les zones pertinentes qu'avec SPE. Mais aussi, les sujets utilisant Dong démontrent une meilleure interprétation des zones d'intérêt que ceux utilisant UML Collaboration Notation ou SPE. Cette observation s'explique par un mélange des informations des patrons et des informations du diagramme de classes chez UML Collaboration Notation et une difficulté à différencier les couleurs chez SPE. **Notre étude confirme donc la conclusion de Cepeda Porras pour cette tâche.**

La représentation UML Collaboration Notation souffre du même problème pour la tâche « rôle ». En effet, les sujets possèdent un pourcentage de transition des zones d'intérêt vers les zones de non-intérêt le plus élevé avec UML Collaboration Notation. Par conséquent, nous pouvons émettre l'hypothèse qu'UML Collaboration Notation souffre d'une mauvaise interprétation de ses zones pertinentes. Cependant, **notre étude nuance les propos de Cepeda Porras pour cette tâche** mais elle ne contredit pas non plus sa conclusion. Cette nuance vient de plusieurs résultats qui ont montrés que les utilisateurs d'UML Collaboration Notation ont une durée moyenne des fixations significativement plus faible que les utilisateurs de Dong. Cette faible valeur signifie qu'UML Collaboration Notation présente les informations des patrons d'une telle façon qu'elles peuvent être rapidement traitées et interprétées par les sujets. Ensuite, les sujets utilisant UML Collaboration Notation possèdent le ratio durée fixation/saccade le plus faible. Ainsi, d'après ces deux conclusions, il semble que la durée du traitement est inférieure chez UML Collaboration Notation.

15.1.3 Question de prospection

L'expérience de Cepeda Porras ayant laissé la porte ouverte quant à l'inexistence d'une représentation idéale pour toutes les tâches, nous nous sommes intéressés à cette question qui se concentre sur la recherche de cette représentation idéale. Il ne s'agit ici que d'une ébauche, nous encourageons d'autres personnes à approfondir celle-ci.

Question 6 : Est-ce qu'une nouvelle représentation peut corriger les lacunes des autres ?

Une représentation utilisant à la fois l'aspect textuel de Dong tout en utilisant plus l'aspect visuel est le compromis à rechercher. On a vu qu'une utilisation intrusive d'un surplus visuel comme utilisé

dans UML Collaboration Notation est handicapant pour les tâches « composition » et « rôle ».

Vu les bon résultats de Dong, l'idée serait de reprendre cette représentation mais de l'améliorer dans la tâche « participation ». Entourer les classes d'une ligne de couleur qui identifie dans quels design patterns elle appartient peut aider la représentation Dong à avoir de meilleurs résultats pour cette tâche. La représentation que nous avons proposée reprend cette idée. Hélas, une utilisation de couleurs non appropriées a donné lieu à des difficultés d'utilisation non attendues... Sur les trois représentations testées, cette représentation arrive souvent deuxième et parfois troisième aux différents résultats. Un résumé des résultats de cette représentation est donné dans le chapitre 14. Il serait donc crucial de penser à sélectionner les couleurs suivant une métrique bien précise. Il est également important de soulever le fait qu'un trait plus épais pour les couleurs faciliterait la différenciation dans le diagramme de classes.

15.2 Autres contributions

Bien que l'étude empirique soit le cœur de notre travail, nous avons contribué à d'autres projets.

Expérimentation

Les résultats significatifs obtenus sont une contribution dans le domaine du génie logiciel. En effet, nous avons pratiquement réalisé l'ensemble des travaux futurs proposés par Porras. De plus, une explication des résultats est apportée alors que Cepeda Porras se contentait d'une énumération de conclusion. Par cette profondeur que nous apportons, nous mettons l'accent via les analyses présentées dans les chapitres 10, 11 et 12 sur les éléments réellement important que doivent avoir une représentation voulant aider à la compréhension de la structure d'un programme contenant des design patterns.

TAUPE

Comme le lecteur peut le remarquer au sein de la section 7.2, nous avons également participé à l'amélioration de ce logiciel développé par le laboratoire PtiDej. La propreté et la documentation de la version 2.0 (proposée par De Smet et Lempereur) nous ont aidé dans ce travail. Notre contribution visant surtout à l'ajout de nouvelles métriques et à l'ajout de certains outils.

Distance relative entre chemins visuels

Enfin, l'étude réalisée l'année dernière par De Smet et Lempereur [DL11] mettait en avant une nouvelle métrique appelée « Distance relative entre chemins visuels ». Cependant, ceux-ci proposaient des évolutions possibles pour cette métrique. Dès lors, nous avons effectué des recherches et apporté l'algorithme *Dynamic Time Warping* à cette métrique qui la rend plus performante.

Plus concrètement, l'utilisation de *Dynamic Time Warping* par rapport à la distance de Levenshtein permet d'éviter des imprécisions dû fait que la distance est calculée par rapport aux fixations et non plus aux zones d'intérêt. Grâce au fait que *Dynamic Time Warping* ne prend plus en compte les zones d'intérêt, ce système permet d'enlever une contrainte. En effet, la limitation qui imposait de ne pas prendre les fixations successives d'une zone d'intérêt n'existe plus. De plus, certains problèmes qui étaient inhérentes à l'utilisation de la distance de Levenshtein ne sont plus présents¹ Enfin, l'algorithme *Dynamic Time Warping* s'applique facilement aux chemins visuels, et est donc optimisé pour une telle recherche.

1. cf. chapitre 8 pour plus d'informations sur ce sujet.

15.3 Travaux futurs

D'une manière globale, nous sommes assez satisfaits des résultats de notre expérimentation. Cependant, cette dernière souffre quand même de quelques limitations. Ceux-ci pourraient donner lieu à de futurs travaux.

Premièrement, l'échec de la nouvelle représentation est assez décevante. Il serait bon de refaire une étude en prenant en compte le facteur complexe que représente l'utilisation des couleurs. Une métrique devrait être définie afin d'identifier les couleurs compatibles. Le but étant surtout que le sujet qui teste la représentation ne soit pas gêné par l'utilisation de celle-ci, mais bien avantagé (ce qui n'était pas le cas ici).

Deuxièmement, notre étude ne se base que sur des représentations statiques. Même si les représentations statiques sont les plus répandues, des études sur des méthodes de visualisation dynamique des informations relatives aux patrons de conception pourraient être envisagées. Cette étude pourrait notamment amener une solution quant à l'utilisation d'une représentation idéale pour les trois tâches.

Troisièmement, un projet qui consisterait à intégrer ces représentations dans des outils de création, d'édition et visualisation de diagrammes de classes pourrait être élaboré. En effet, malgré le fait que nous avons conclu que certaines représentations étaient meilleurs que d'autres, rien n'encourage les développeurs à utiliser celles-ci. Par contre, la création d'un outil proposant l'utilisation des ces différents outils pourraient apporter une innovation dans ce domaine.

Enfin, une remarque venant de Wholin *et al.* [WRH+00] (notre expérience se réfère à leur définition), n'a pas été prise en compte. Concrètement, les auteurs disent qu'il est préférable d'analyser l'objet d'étude dans son environnement afin d'assurer des résultats généraux les plus valides possibles. Dans notre cas, il s'agirait de comprendre comment une représentation peut influencer la compréhension d'une structure d'un programme industriel utilisé par des professionnels. En effet, nous rappelons que l'entièreté des sujets de notre étude sont des étudiants en informatique. Cette remarque, bien que difficile à appliquer, serait intéressante à prendre en compte dans une prochaine expérimentation.

Bibliographie

- [AGJ01] Albin-Amiot, H., Cointe, P., Guéhéneuc, Y-G. et Jussien, N., *Instantiating and Detecting Design Patterns : Putting Bits and Pieces Together*, 2001.
- [AIS+77] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, et Shlomo A., *A Pattern Language*, Oxford University Press, New York, 1977.
- [Amb04] Ambler, S. W., *The Object Primer : Agile Model-Driven Development with UML 2.0*. Cambridge University Press, 2004.
- [BCR94] Basili, V. R., Caldiera, G., et Rombach, H. D. The goal question metric approach. In *Encyclopedia of Software Engineering*, Wiley, 1994.
- [BG97] Bellay, B. et Gall, H., A comparison of four reverse engineering tools, In Ira BAXTER et Alex QUILICI, éditeurs : *Proceedings of the 4th Working Conference on Reverse Engineering*, pages 2–11, IEEE Computer Society Press, octobre 1997.
- [BMR+96] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., et Stal, M., *Pattern-Oriented Software Architecture, Volume 1 : A System of Patterns*, Wiley, Chichester, UK, 1996.
- [BT04] Bednarik, R. et Tukiainen, M., *Visual attention tracking during program debugging*, pages 331–334, ACM, New York, USA, 2004.
- [BT06] Bednarik, R. et Tukiainen, M., An eye-tracking methodology for characterizing program comprehension processes, In *Proceedings of 5th symposium on Eye Tracking Research & Applications*, pages 125–132, ACM Press, 2006.
- [Bro78] Brooks, R., *Using a behavioral theory of program comprehension in software engineering*. IEEE Press, Piscataway, NJ, USA, 1978.
- [CK05] Chabris, C. F. et Kosslyn, S. M., Representational correspondence as a basic principle of diagram design, In *Knowledge and Information Visualization*, pages 36-57, SpringerVerlag, 2005.
- [Cai07] Cain, B., *A Review of the Mental Workload Literature*. Defence Research and Development Canada Toronto. juillet 2007.
- [DL11] De Smet, B. et Lempereur L., *Étude empirique de l'impact du patron architectural MVC et de ses variantes sur la maintenabilité interne des programmes*, Mémoire de master, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgique, 2011.
- [DPA+92] Dewayne E. Perry, Alexander L. Wolf., Foundations for the Study of Software Architecture, *ACM SIGSOFT Software Engineering Notes*, vol 17, no 4, page 40, oct 1992.
- [DYZ07] Dong J., Yang S. and Zhang K., *Visualizing design patterns in their applications and compositions*. IEEE Transactions on Software Engineering, 2007.

- [Dre10] Drewes, H., *Eye Gaze Tracking for Human Computer Interaction*. Thèse de doctorat en sciences des média informatique, Université Ludwig-Maximilians de Munich, 2010.
- [Duc07] Duchowski, A. T., *Eye tracking methodology. Theory and Practice*, 2007.
- [GHJ+94] Gamma, E., Helm, R., Johnson, R., et Vlissides, J. M., *Design Patterns : Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994.
- [GK99] Goldberg, H. J. et Kotval, X. P., *Computer interface evaluation using eye movements : Methods and constructs*, *International Journal of Industrial Ergonomics*, 24, 631-645, 1999.
- [GSZ04] Guéhéneuc, Y-G., Sahraoui, H., et Zaidi, F., *Fingerprinting Design Patterns*, 2004.
- [Gam96] Gamma, E., *Applying design patterns in Java*, *Java Report*, 1(6) :47-53, 1996.
- [Glass94] Glass, R., *The Software Research Crisis*, *IEEE Software*, pg 42-47, Nov. 1994.
- [Gus97] Gusfield, D., *Algorithms on strings, trees, and sequences : Computer science and computational biology.*, Cambridge University Press, New York, USA, 1997.
- [Gué06] Guéhéneuc, Y.-G., *Taupe : Towards understanding program comprehension.*, In Hakan ERDOGMUS et Eleni STROULIA, éditeurs : *Proceedings of the 16th IBM Centers for Advanced Studies Conference (CASCON)*, pages 1-13. ACM Press, octobre 2006.
- [Gué08] Guéhéneuc Y., *A Theory of Program Comprehension Joining Vision Science and Program Comprehension*. Montreal, Canada, October 2008.
- [HS88] Hart, S. et Staveland, L., *Development of NASA-TLX (Task Load Index) : Results of empirical and theoretical research*, *Human Mental Workload*, 1988.
- [HS88] Hart SG, Staveland LE., *Development of NASA-TLX (Task Load Index) : results of empirical and theoretical research*. En : Hancock PA, Meshkati N, eds. *Human Mental Workload.*, Amsterdam : Elsevier, 1988.
- [IEE90] IEEE. *Ieee standard glossary of software engineering terminology*, IEEE Std 610.12-1990, 1990.
- [JC76] Just, M. A., et Carpenter, P. A., *Eye fixations and cognitive processes*, *Cognitive Psychology*, 8, 441-480, 1976.
- [JM10] Juristo, N. et Moreno, A. M., *Basics of Software Engineering Experimentation*, Springer Publishing Company, Incorporated, 1st édition, 2010.
- [Jean08] Jeanmart, S., *Évaluation de l'impact d'un patron de conception sur la compréhension et la maintenance de programmes – une experimentation par un système d'eye-tracking*, Mémoire de master, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgique, 2008.
- [Jim07] Jiménez, J. M. N. *Cours Ergonomie cognitive : Charge mentale de travail.*, École Polytechnique de Montréal, Décembre 2007.
- [KC87] B. Kent et C. Cunningham, *Using Pattern Languages for Object-Oriented Program, Specification and Design for Object-Oriented Programming*, 1987.
- [KKB+98] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lispon et J. Carriere, *In ICECCS '98 : Proceedings of the Fourth IEEE International Conference on Engineering of Complex Computer Systems*, IEEE Computer Society, 1998.
- [lar12] Larousse. Identificateur — Larousse, Dictionnaire. <http://www.larousse.fr/dictionnaires/francais/identificateur/41411>, août 2012. Récupéré le 24 août 2012.

- [MS88] Marascuilo, L. A. et Serlin, R. C., *Statistical Methods for the Social and Behavioral Sciences*, W. H. Freeman and Company, New York, USA, 1988.
- [Mal08] Jonathan I. Maletic, *Ieee international conference on program comprehension (icpc)*, Avril 2008.
- [Mar95] Mayrhauser A., *Program comprehension during software maintenance and evolution*, Colorado State university, August 1995.
- [Nie04] Niels, R., *Dynamic Time Warping : An intuitive way of handwriting recognition ?*, Mémoire de master, Radboud University Nijmegen, Nijmegen, Pays-Bas, 2004.
- [Niels04] Niels, R., *Dynamic Time Warping : An intuitive way of handwriting recognition ?*, University Nijmegen, December 2004.
- [PB06] Poole, A. et Ball, L. J., *Eye tracking in Human-Computer interaction and usability research : Current status and future prospects.*, Ghaoui, Claude (Ed.), *Encyclopedia of Human Computer Interaction*, 2006.
- [PBP04] Poole, A., Ball, L. J., et Phillips, P., *In search of salience : A response time and eye movement analysis of bookmark recognition*, *People and Computers XVIII-Design for Life : Proceedings of HCI 2004*, London : Springer-Verlag Ltd, 2004.
- [PUP+02] Prechelt, L., Unger-Lamprecht, B., Philippsen, M., et Tichy, W. F., *Two controlled experiments assessing the usefulness of design pattern documentation in program maintenance*, *IEEE Trans. Softw. Eng.*, 28 :595–606, juin 2002.
- [PUT+01] Prechelt, L., Unger, B., Tichy, W., Brossler, P., et Votta, L., *A controlled experiment in maintenance : comparing design patterns to simpler solutions*, *Software Engineering*, *IEEE Transactions on*, 27(12) :1134–1144, December 2001.
- [Pal99] Palmer, S. E., *Vision Science : Photons to Phenomenology*, The MIT Press, 1st édition, mai 1999.
- [Por08] Porras G., *Analyse, à l'aide d'oculomètres, de techniques de visualisation UML de patrons de conception pour la compréhension de programmes*, Mémoire de master, Université de Montreal, Montreal, QC, Canada, 2008.
- [RC07] Rayner, K. et Castelhana, M., *Eye Movements*, http://www.scholarpedia.org/article/Eye_movements, Scholarpedia, 2007. Page consultée le 8 avril 2012.
- [RL01] Redline, C. D., et Lankford, C. P., *Eye-movement analysis : A new tool for evaluating the design of visually administered instruments (Paper and Web)*, *Proceedings of the Section on Survey Research Methods of the American Statistical Association*, 2001.
- [RT05] Richeter, A. et Ternaux M., *Les mouvements oculaires*, <http://acces.ens-lyon.fr/acces/ressources/neurosciences/vision/VisionMarseille>, 2005-2006, Page consultée le 8 avril 2012.
- [Ray98] Rayner, K., *Eye Movements in Reading and Information Processing : 20 Years of Research*, *Psychological bulletin*, 1998.
- [Rug95] Rugaber S., *Program Comprehension*. Georgia Institute of Technology, May 1995.
- [SC88] Siegel, S. et Castellan, J., *Nonparametric Statistics for the Behavioral Sciences*, 2^e édition, McGraw-Hill International Editions, 1988.
- [SFM99] Storey, M.-A. D., Fracchia, F. D., et Müller, H. A., *Cognitive design elements to support the construction of a mental model during software exploration*, *Journal of Systems and Software*, 44(3) :171–185, janvier 1999.

- [SK98] Schauer R. et Keller R., Pattern visualization for software comprehension. In *IWPC '98 : Proceedings of the 6th International Workshop on Program Comprehension*, pages 4–12, IEEE Computer Society, 1998.
- [SM10] Sharif, B. et Maletic, J. I., *An eye tracking study on camelCase and under_score identifier styles*, International Conference on Program Comprehension, 0 :196–205, Juin–Juillet 2010.
- [Sta05] Stanton, N., *Human factors methods : A practical guide for engineering and design*, Ashgate Pub. Co., 2005.
- [Sto06] Storey M., *Theories, Methods and Tools in Program Comprehension : Past, Present and Future*, University of Victoria, Canada, 2006.
- [TCS+06] Tsantalis, N., Chatzigeorgiou, A., Stephanides, G. et T. Halkidis Spyros., *Design pattern detection using similarity scoring*, 2006.
- [TWK+10] Tatler, B. W., Wade, N. J., Kwan, H., Findlay, J. M. et Velichkovsky, B. M., *Yarbus, eye movements, and vision, i-perception*, volume 1, July 2010.
- [VAL+00] V. Vuori, M. Aksela, J. Laaksonen, E. Oja, et J. Kangas., Adaptive character recognizer for a hand-held device : Implementation and evaluation setup, In *Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, Netherlands, Septembre 2000.
- [Vdp09] Van Den Plas, B., *La theorie Vision-Comprehension appliquée aux patrons de conception*. Mémoire de master, Facultes Universitaires Notre-Dame de la Paix, Namur, Belgique, 2009.
- [Vli98] Vlissides J., *Notation, notation, notation*, C++ Report, April 1998.
- [WF08] Wilkinson L. et Friendly M., *The History of the Cluster Heat Map*, The American Statistician, November 2008
- [WF74] Wagner, R. A. et Fischer, M. J., *The string-to-string correction problem*, January 1974.
- [WRH+00] Wholin, C., Runeson, P., Ohlsson, M. C., Regnell, B. et Wesslén, A., *Experimentation in Software Engineering - An Introduction*, Kluwer Academic Publishers, 1st édition, 2000.
- [Wen01] Wendorff, P., Assessment of design patterns during software reengineering : Lessons learned from a large commercial project, In *Proceedings of the Fifth European Conference on Software Maintenance and Reengineering*, CSMR '01, pages 77–, Washington, DC, USA, 2001.
- [YKM07] Yusuf, S., Kagdi, H., et Maletic, J. I., Assessing the comprehension of UML class diagrams via eye tracking, In Eleni STROULIA et Paolo TONELLA, éditeurs : *Proceedings of the 15th International Conference on Program Comprehension*, pages 113–122, IEEE Computer Society Press, juin 2007.
- [Yar67] Yarbus A. L., *Eye Movements and Vision*, New York : Plenum Press, 1967.
- [vMV95] von Mayrhauser, A. et Vans, A. M., *Program Comprehension During Software Maintenance and Evolution*, volume 28, pages 44–55, IEEE Computer Society Press, Los Alamitos, CA, USA, 1995.
- [wiki12] Wikipedia. Framework — Wikipedia, L'encyclopédie libre. <http://fr.wikipedia.org/wiki/Framework>, Août 2012. Récupéré le 24 Août 2012.

Tableaux de données

A.1 Tableau de répartition sujet-représentation-framework

Sujet	Rep	Framework	Sexe	Niveau d'étude
S01	UMLN	JRefactory	M	Doctorant
S01	Dong	Nutch	M	Doctorant
S02	Dong	JRefactory	M	Master
S02	SPE	Nutch	M	Master
S03	SPE	JRefactory	M	Master
S03	UMLN	Nutch	M	Master
S04	Dong	JRefactory	F	Master
S04	UMLN	Nutch	F	Master
S05	SPE	JRefactory	F	Doctorant
S05	Dong	Nutch	F	Doctorant
S06	UMLN	JRefactory	F	Doctorant
S06	SPE	Nutch	F	Doctorant
S07	UMLN	JRefactory	M	Master
S07	Dong	Nutch	M	Master
S08	Dong	JRefactory	F	Doctorant
S08	SPE	Nutch	F	Doctorant
S09	SPE	JRefactory	M	Master
S09	UMLN	Nutch	M	Master
S10	Dong	JRefactory	M	Master
S10	UMLN	Nutch	M	Master
S13	UMLN	JRefactory	F	Doctorant
S13	Dong	Nutch	F	Doctorant
S14	Dong	JRefactory	F	Doctorant
S14	SPE	Nutch	F	Doctorant
S15	SPE	JRefactory	M	Doctorant
S15	UMLN	Nutch	M	Doctorant
S16	Dong	JRefactory	M	Master
S16	UMLN	Nutch	M	Master
S18	UMLN	JRefactory	M	Doctorant
S18	SPE	Nutch	M	Doctorant

S19	UMLN	JRefactory	M	Doctorant
S19	Dong	Nutch	M	Doctorant
S20	SPE	JRefactory	M	Master
S20	UMLN	Nutch	M	Master

Tableau A.1 – Répartition sujet-représentation-framework

A.2 Tableau des mesures de performances

Sujet	Rep	Question	Temps passé sur la tâche
S01	UMLNotation	C1	69562
S01	UMLNotation	C2	97317
S01	UMLNotation	P1	76031
S01	UMLNotation	P2	38241
S01	UMLNotation	R1	34515
S01	UMLNotation	R2	49835
S01	Dong	C1	119579
S01	Dong	C2	98719
S01	Dong	P1	51093
S01	Dong	R1	32328
S01	Dong	R2	19313
S02	Dong	C1	43699
S02	Dong	C2	19756
S02	Dong	P1	57664
S02	Dong	P2	35892
S02	Dong	R1	21934
S02	Dong	R2	22845
S02	SPE	C1	23719
S02	SPE	C2	20389
S02	SPE	P1	32921
S02	SPE	R1	12513
S02	SPE	R2	20140
S03	SPE	C2	41008
S03	SPE	P2	24442
S03	SPE	R2	22050
S03	UMLNotation	C1	45125
S03	UMLNotation	C2	21306
S03	UMLNotation	P1	31576
S03	UMLNotation	P2	38146
S03	UMLNotation	R1	26016
S03	UMLNotation	R2	79023
S04	Dong	C1	21645
S04	Dong	C2	16645

Sujet	Rep	Question	Temps passé sur la tâche
S04	Dong	P1	34289
S04	Dong	P2	36956
S04	Dong	R1	12430
S04	Dong	R2	13200
S04	UMLNotation	C1	30053
S04	UMLNotation	C2	26826
S04	UMLNotation	P1	15796
S04	UMLNotation	P2	10724
S04	UMLNotation	R1	23741
S04	UMLNotation	R2	30532
S05	SPE	C1	34010
S05	SPE	C2	35473
S05	SPE	R1	15734
S05	SPE	R2	19089
S05	Dong	C1	70296
S05	Dong	C2	17647
S05	Dong	R1	17468
S05	Dong	R2	35266
S06	UMLNotation	C1	39876
S06	UMLNotation	C2	18765
S06	UMLNotation	P2	16333
S06	UMLNotation	R1	18566
S06	UMLNotation	R2	34669
S06	SPE	C1	58111
S06	SPE	C2	14326
S06	SPE	P1	25258
S06	SPE	R1	30286
S06	SPE	R2	30659
S07	UMLNotation	C1	20224
S07	UMLNotation	C2	18332
S07	UMLNotation	P1	13746
S07	UMLNotation	P2	12774
S07	UMLNotation	R1	9750
S07	UMLNotation	R2	17449
S07	Dong	C1	18298
S07	Dong	C2	14842
S07	Dong	P1	55697
S07	Dong	P2	43928
S07	Dong	R1	10387
S07	Dong	R2	16467
S08	Dong	C1	28571
S08	Dong	C2	27095
S08	Dong	P1	43611
S08	Dong	P2	29977

Sujet	Rep	Question	Temps passé sur la tâche
S08	Dong	R1	23177
S08	Dong	R2	17726
S08	SPE	C1	19436
S08	SPE	C2	22075
S08	SPE	P1	16110
S08	SPE	R1	7966
S08	SPE	R2	9251
S09	SPE	C1	14800
S09	SPE	C2	17843
S09	SPE	P1	56086
S09	SPE	R1	13509
S09	SPE	R2	11566
S09	UMLNotation	C1	15408
S09	UMLNotation	C2	8088
S09	UMLNotation	P1	7012
S09	UMLNotation	R2	21781
S10	Dong	C1	17655
S10	Dong	C2	10562
S10	Dong	P1	16524
S10	Dong	P2	16560
S10	Dong	R1	10073
S10	Dong	R2	10534
S10	UMLNotation	C2	15752
S10	UMLNotation	P1	19349
S10	UMLNotation	P2	11310
S10	UMLNotation	R1	5988
S10	UMLNotation	R2	10398
S13	UMLNotation	C1	23186
S13	UMLNotation	C2	39848
S13	UMLNotation	P2	67557
S13	UMLNotation	R1	7792
S13	Dong	C1	15395
S13	Dong	C2	16270
S13	Dong	P1	16481
S13	Dong	R1	17520
S13	Dong	R2	21257
S14	Dong	C1	22479
S14	Dong	C2	14020
S14	Dong	P2	38819
S14	Dong	R1	14136
S14	Dong	R2	17863
S14	SPE	C1	86267
S14	SPE	C2	9631
S14	SPE	P1	47933

Sujet	Rep	Question	Temps passé sur la tâche
S14	SPE	R1	18307
S14	SPE	R2	10790
S15	SPE	C1	21369
S15	SPE	C2	60285
S15	SPE	P1	57720
S15	SPE	P2	87539
S15	SPE	R1	24687
S15	SPE	R2	50694
S15	UMLNotation	C1	45731
S15	UMLNotation	C2	54093
S15	UMLNotation	P1	28066
S15	UMLNotation	P2	51333
S15	UMLNotation	R2	25540
S16	Dong	C2	40384
S16	Dong	P1	54495
S16	UMLNotation	C1	30595
S16	UMLNotation	C2	30963
S16	UMLNotation	P2	15654
S18	UMLNotation	C1	11556
S18	UMLNotation	C2	12299
S18	UMLNotation	P1	17210
S18	UMLNotation	P2	9615
S18	UMLNotation	R1	10683
S18	SPE	C1	38563
S18	SPE	C2	18270
S18	SPE	P1	21436
S18	SPE	P2	11328
S18	SPE	R1	16181
S18	SPE	R2	16719
S19	UMLNotation	C1	46812
S19	UMLNotation	C2	62946
S19	UMLNotation	P1	57000
S19	UMLNotation	P2	21828
S19	UMLNotation	R1	36131
S19	UMLNotation	R2	60093
S19	Dong	C2	24239
S19	Dong	R1	47035
S19	Dong	R2	17162
S20	SPE	C1	95156
S20	SPE	C2	30493
S20	SPE	P2	13209
S20	SPE	R1	7767
S20	SPE	R2	15331
S20	UMLNotation	C1	67937

Sujet	Rep	Question	Temps passé sur la tâche
S20	UMLNotation	C2	29408
S20	UMLNotation	P1	27369
S20	UMLNotation	P2	15681
S20	UMLNotation	R2	29624

Tableau A.2 – Données de performance

A.3 Tableaux des mesures physiologiques

A.3.1 Tableau partie « question »

Sujet	Rep	Question	DS	DT	IA/IAN	DMF	RDF/S	PTAVA	NMFPA
S01	UMLN	C1	10,196	0,0969	0,9088	77,7692	0,1598	4,4776	840,0
S01	UMLN	C2	19,0196	0,0754	0,8073	74,5809	0,1609	4,9382	112,6667
S01	UMLN	P1	7,8431	0,0838	0,8605	87,0593	0,1351	6,25	64,25
S01	UMLN	P2	8,2352	0,0203	0,6656	72,6250	0,1671	1,6666	410,0
S01	UMLN	R1	5,8823	0,0011	0,9615	78,7093	0,1961	9,3023	830,0
S01	UMLN	R2	7,6471	0,0659	0,8049	78,2929	0,1555	7,8125	800,0
S01	Dong	C1	19,6078	0,1999	0,6197	75,0418	0,2404	3,3898	2790,0
S01	Dong	C2	15,8823	0,0718	0,8936	80,1381	0,2469	3,4884	2030,0
S01	Dong	P1	11,7647	0,0814	0,4423	79,3478	0,2858	2,924	510,0
S01	Dong	R1	4,50980	0,0562	0,9931	84,1129	0,3228	2,5862	1040,0
S01	Dong	R2	5,4902	0,0538	0,9192	77,0164	0,2434	3,9216	210,0
S02	Dong	C1	10	0,0778	0,9338	84,0988	0,3118	3,2895	1180,0
S02	Dong	C2	4,1176	0,018	0,9633	90,2292	0,2194	2,3256	110,0
S02	Dong	P1	11,7647	0,1042	0,8619	76,8082	0,2917	3,3654	168,8
S02	Dong	P2	8,8235	0,0359	0,6388	77,9333	0,2606	4,5045	87,4286
S02	Dong	R1	3,1373	0,0048	0,9877	82,0896	0,2508	3,2787	610,0
S02	Dong	R2	4,3137	0,0347	0,9871	80,2593	0,2846	2,7027	420,0
S02	SPE	C1	5,0980	0,0371	0,9146	76,7797	0,191	3,7735	230,0
S02	SPE	C2	4,7059	0,0168	0,9118	67,7344	0,2126	3,2258	150,0
S02	SPE	P1	6,8627	0,0323	0,9386	66,4595	0,1494	5,7971	53,2857
S02	SPE	R1	2,1569	0,0251	0,9887	75,4906	0,3201	2	430,0
S02	SPE	R2	3,1373	0,0395	0,9928	830,0	0,2184	6,25	470,0
S03	SPE	C2	6,4705	0,0251	0,7339	72,0357	0,0983	4,5454	7,5
S03	SPE	P2	5,8823	0,0239	0,4839	85,5882	0,1785	4,6511	20,625
S03	SPE	R2	4,9019	0,0526	0,9589	73,36	0,2499	4,4117	510,0
S03	UMLN	C1	9,4117	0,0814	0,9264	71,6049	0,1285	10,0	51,4
S03	UMLN	C2	3,3333	0,0011	0,9603	70,4516	0,1025	11,1111	100,0
S03	UMLN	P1	3,7254	0,0371	0,9770	62,4615	0,1028	2,5	18,2142
S03	UMLN	P2	6,4705	0,0095	0,9916	760,0	0,1734	1,5384	42,7333
S03	UMLN	R1	5,8823	0,0359	0,9021	77,1525	0,1750	10,0	570,0
S03	UMLN	R2	10	0,1005	0,9610	72,3233	0,1217	7,8947	1270,0
S04	Dong	C1	3,1372	0,0275	0,9951	82,6756	0,2837	0,0	48
S04	Dong	C2	2,3529	0,0071	0,9868	73,2333	0,1320	0,0	6
S04	Dong	P1	8,8235	0,0502	0,5194	76,0693	0,2240	1,0638	91,6
S04	Dong	P2	9,0196	0,0610	0,3487	74,6964	0,2264	0,9615	41,1428
S04	Dong	R1	4,3137	0,0011	0,9670	86,9047	0,2957	2,7777	27
S04	Dong	R2	1,9607	0,0203	0,9951	77,4583	0,2835	0,0	43
S04	UMLN	C1	4,5098	0,0490	0,9707	84,8750	0,1807	9,0909	24
S04	UMLN	C2	5,8823	0,0179	0,8732	73,2641	0,1448	0,0	9
S04	UMLN	P1	4,1176	0,0335	1	77,3333	0,2072	0,0	11,2857
S04	UMLN	P2	2,3529	0,0023	1	68,5882	0,1095	0,0	3,5333

Sujet	Rep	Question	DS	DT	IA/IAN	DMF	RDF/S	PTAVA	NMFPA
S04	UMLN	R1	4,3137	0,0323	0,9448	74,9493	0,2502	1,5151	29,0
S04	UMLN	R2	3,9215	0,0431	0,9875	75,5128	0,2893	0,0	64,0
S05	SPE	C1	6,6666	0,0550	0,9363	80,9532	0,2547	2,9411	71,0
S05	SPE	C2	5,2941	0,0095	0,8643	74,7187	0,1348	9,6774	48,5
S05	SPE	R1	1,9607	0,0	0,9982	85,0357	0,3027	0,0	0,0
S05	SPE	R2	2,7450	0,0287	0,9851	79,7297	0,1552	0,0	0,0
S05	Dong	C1	11,9607	0,1245	0,8642	76,1659	0,2676	0,0	64,0
S05	Dong	C2	3,7254	0,0155	0,9475	80,7590	0,3798	2,4691	54,0
S05	Dong	R1	2,9411	0,0395	0,9949	72,4058	0,2860	0,0	0,0
S05	Dong	R2	4,7058	0,0431	0,9268	75,70492	0,13094	3,3333	34,0
S06	UMLN	C1	9,6078	0,0909	0,7425	80,2777	0,2174	1,1111	130,0
S06	UMLN	C2	6,4705	0,0239	0,9537	69,9032	0,2309	8,8888	40,3333
S06	UMLN	P2	5,2941	0,0011	0,9885	82,1090	0,2764	0,0	23,8
S06	UMLN	R1	6,4705	0,0155	0,8413	70,7077	0,2475	3,7037	450,0
S06	UMLN	R2	7,0588	0,0790	0,9724	71,7631	0,2359	7,5471	1030,0
S06	SPE	C1	11,9607	0,0838	0,8824	89,1923	0,2394	4,0816	105,5
S06	SPE	C2	3,5294	0,00838	0,9268	80,1698	0,2968	0,0	00,0
S06	SPE	P1	10,9803	0,0550	0,8341	73,6393	0,3558	5,8252	920,0
S06	SPE	R1	7,6470	0,0514	0,9235	87,1311	0,3509	1,7543	640,0
S06	SPE	R2	8,0392	0,0658	0,9170	76,0277	0,2679	2,0618	430,0
S07	UMLN	C1	6,2745	0,0371	0,7821	72,0535	0,1995	9,0909	28,6
S07	UMLN	C2	4,7058	0,0107	0,9748	75,6046	0,1774	10,0	19,3333
S07	UMLN	P1	2,3529	0,0179	0,8621	71,65	0,1043	5,8823	4,25
S07	UMLN	P2	2,7450	0,0035	0,4117	82,35	0,1289	17,6470	3,4666
S07	UMLN	R1	1,1764	0,0	0,9286	65,7058	0,1145	12,5	10,0
S07	UMLN	R2	3,7254	0,0275	0,9749	60,7058	0,1183	8	18
S07	Dong	C1	2,9411	0,0335	0,9621	71,5344	0,2269	3,5714	44,0
S07	Dong	C2	4,1176	0,0095	0,9402	72,2222	0,1751	3,0303	13,0
S07	Dong	P1	12,5490	0,0838	0,4873	77,7687	0,2052	6,0150	95,50
S07	Dong	P2	9,4117	0,0466	0,7210	77,1351	0,1949	4,8543	58,8571
S07	Dong	R1	1,9607	0,0275	0,9918	860,0	0,3644	0,0	8,0
S07	Dong	R2	2,7450	0,0095	0,8661	71,6666	0,1175	0,0	0,0
S08	Dong	C1	4,7058	0,05028	0,9731	90,2906	0,3698	0,0	00,0
S08	Dong	C2	6,6666	0,0227	0,8999	79,7681	0,2032	3,3333	270,0
S08	Dong	P1	11,1764	0,0862	0,7787	83,1505	0,3547	1,0989	117,2
S08	Dong	P2	9,0196	0,0466	0,3833	81,1875	0,3034	2,8037	690,0
S08	Dong	R1	5,4901	0,0275	0,9444	80,7666	0,3137	0,0	00,0
S08	Dong	R2	4,9019	0,0383	0,9294	77,5918	0,2146	0,0	30,0
S08	SPE	C1	4,5098	0,0395	0,8101	93,5	0,2983	1,6949	60,0
S08	SPE	C2	5,0980	0,0239	0,9135	67,7808	0,2250	2,8571	160,0
S08	SPE	P1	4,1176	0,0131	0,9140	73,1136	0,1998	4,6511	12,8571
S08	SPE	R1	1,7647	0,0203	1	88,3250	0,4439	0,0	390,0
S08	SPE	R2	1,9607	0,0203	0,9584	76,8620	0,2411	0,0	00,0
S09	SPE	C1	2,3529	0,0179	0,8988	71,9545	0,1069	5,8823	8,50

Sujet	Rep	Question	DS	DT	IA/IAN	DMF	RDF/S	PTAVA	NMFPA
S09	SPE	C2	6,6666	0,0167	0,6315	70,6304	0,1821	2,7027	4,0
S09	SPE	P1	11,5686	0,0574	0,5943	74,3697	0,1577	2,8301	70,8333
S09	SPE	R1	3,5294	0,0107	0,9583	94,2894	0,2652	2,8571	19,0
S09	SPE	R2	1,9607	0,0155	0,9755	81,9615	0,1843	4,1666	21,0
S09	UMLN	C1	3,9215	0,0251	0,9118	71,5161	0,1439	8,0	7,6
S09	UMLN	C2	2,3529	0,0059	0,9785	80,3529	0,1688	0,0	1,6666
S09	UMLN	P1	2,5490	0,0131	1	83,3333	0,2142	0,0	4,42857
S09	UMLN	R2	4,5098	0,0275	0,9574	79,15	0,1454	9,0909	24
S10	Dong	C1	4,3137	0,0407	0,9787	74,3278	0,2568	3,8461	56
S10	Dong	C2	1,7647	0,0059	0,9727	79,0714	0,2097	0,0	0,0
S10	Dong	P1	4,7058	0,0203	0,8765	81,1666	0,2072	5,2631	31
S10	Dong	P2	4,5098	0,0131	0,5069	69,1818	0,1378	6,6666	11,5714
S10	Dong	R1	1,9607	0,0011	0,9929	87,5161	0,2695	0,0	0,0
S10	Dong	R2	3,3333	0,0191	0,9072	61,32	0,14566705	4,7619	8
S10	UMLN	C2	2,9411	0,0023	0,9943	87,3103	0,1607	0,0	16,6666
S10	UMLN	P1	2,9411	0,0155	0,8787	80,3823	0,1412	0,0	12,3571
S10	UMLN	P2	2,3529	0,0011	0,9639	81,8095	0,1519	8,3333	3,3333
S10	UMLN	R1	0,9803	0,0059	1	74,8333	0,0759	0,0	0,0
S10	UMLN	R2	1,7647	0,0107	0,9769	67,7647	0,1109	0,0	8
S13	UMLN	C1	2,7450	0,0407	0,9821	80,4843	0,2221	0,0	18,6
S13	UMLN	C2	9,4117	0,0419	0,8194	78,4117	0,3010	0,7575	170,0
S13	UMLN	P2	4,1176	0,0083	1	78,8292	0,0478	0,0	13,4
S13	UMLN	R1	1,5686	0,0	1	74,4666	0,1435	0,0	50,0
S13	Dong	C1	3,3333	0,0227	0,7608	76,1515	0,1633	3,2258	40,0
S13	Dong	C2	3,5294	0,01915	0,942	73,5681	0,1990	0,0	150,0
S13	Dong	P1	3,3333	0,03591	0,374	75,8750	0,2580	3,7735	10,8333
S13	Dong	R1	4,5098	0,02274	0,941	98,85	0,3386	1,7857	250,0
S13	Dong	R2	3,7254	0,02514	0,489	70,1176	0,1122	0,0	00,0
S14	Dong	C1	2,9411	0,0371	0,8715	83,7894	0,2834	5,6338	730,0
S14	Dong	C2	3,3333	0,0083	0,9309	76,5625	0,1748	3,3333	70,0
S14	Dong	P2	7,4509	0,0383	0,4454	71,3066	0,1377	3,2258	18,2857
S14	Dong	R1	2,5490	0,0047	0,9440	79,9666	0,1697	3,5714	240,0
S14	Dong	R2	3,7254	0,0287	0,9517	76,9411	0,1464	3,3333	170,0
S14	SPE	C1	14,3137	0,0957	0,8769	70,2065	0,1497	9,7902	152,5
S14	SPE	C2	2,3529	0,0059	0,9103	66,68	0,1733	4,3478	50,0
S14	SPE	P1	10,7843	0,0395	0,7076	73,3608	0,1484	12,9870	51,1428
S14	SPE	R1	5,2941	0,0275	0,9160	68,4814	0,2020	2	250,0
S14	SPE	R2	2,3529	0,0191	0,9521	77,2272	0,1575	10,0	70,0
S15	SPE	C1	3,9215	0,0395	0,9713	79,9333	0,2244	5,8823	23,5
S15	SPE	C2	10,7843	0,0167	0,8426	71,6635	0,1272	11,1111	830,0
S15	SPE	P1	10,9803	0,0730	0,7541	83,9694	0,1906	5,7851	108,5
S15	SPE	P2	15,2941	0,0371	0,8536	76,0806	0,2911	2,2292	295,75
S15	SPE	R1	5,4901	0,0011	0,9568	80,7435	0,25511	5,4794	480,0
S15	SPE	R2	8,4313	0,0921	0,9497	73,3542	0,25322	3,2258	980,0

Sujet	Rep	Question	DS	DT	IA/IAN	DMF	RDF/S	PTAVA	NMFP
S15	UMLN	C1	8,2352	0,0945	0,9320	740,0	0,1731	8,9743	63,8
S15	UMLN	C2	10	0,0167	0,9061	69,3503	0,17568	8,7378	700,0
S15	UMLN	P1	6,0784	0,0466	0,9644	86,9841	0,1952	1,8867	30,42857
S15	UMLN	P2	9,0196	0,0203	0,9078	81,4080	0,1983	2,1739	50,6
S15	UMLN	R2	4,7058	0,0323	0,9319	82,3181	0,2128	11,9047	550,0
S16	Dong	C2	5,4901	0,0119	0,9762710	78,4622	0,2063	0,99	450,0
S16	Dong	P1	12,3529	0,0921	0,8566698	76,9766	0,2416	2,5157	124,2
S16	UMLN	C1	6,6666	0,055	0,8774	70,2666	0,1378	13,5135	36,2
S16	UMLN	C2	6,6666	0,0107	0,9379	69,6428	0,2520	7,0422	62,3333
S16	UMLN	P2	4,1176	0,0059	0,9705	84,1025	0,2095	0,0	13,0666
S18	UMLN	C1	3,7254	0,0287	0,8658	82,7894	0,2722	9,677420	16,4
S18	UMLN	C2	3,7254	0,0143	0,9096	66,7096	0,1681	8	6
S18	UMLN	P1	3,3333	0,0275	0,9628	98,1111	0,1539	9,0909	8,1666
S18	UMLN	P2	2,5490	0,0011	0,7841	710,0	0,1403	0,0	3,7333
S18	UMLN	R1	4,7058	0,0083	0,8731	71,6969	0,2214	7,1428	21
S18	SPE	C1	7,0588	0,0395	0,9671	83,8507	0,2913	3,125	97
S18	SPE	C2	5,0980	0,0215	0,7211	81,4528	0,2362	1,9607	25
S18	SPE	P1	7,6470	0,0347	0,7503	67,7341	0,2496	8,1081	47,1428
S18	SPE	P2	3,7254	0,0179	0,8196	86,3	0,3830	8,5106	17,125
S18	SPE	R1	2,1568	0,0251	0,9625	83,7808	0,3779	0,0	0,0
S18	SPE	R2	5,0980	0,0299	0,8870	74,4594	0,3305	0,0	0,0
S19	UMLN	C1	9,019	0,0670	0,8531	70,6203	0,1629	6,5934	51,8
S19	UMLN	C2	14,9019	0,0658	0,8221	73,8174	0,1477	4,6728	26,3333
S19	UMLN	P1	7,6470	0,0838	0,8945	75,0795	0,1161	11,1111	45,75
S19	UMLN	P2	5,0980	0,0071	0,8004	69,9130	0,1473	6,9767	23,8666
S19	UMLN	R1	7,6470	0,0071	0,8354	75,6111	0,1506	4,7619	370,0
S19	UMLN	R2	8,0392	0,0514	0,7878	64,6071	0,0602	6,6666	280,0
S19	Dong	C2	4,7058	0,01317	0,9687	73,5416	0,1457	4,5454	200,0
S19	Dong	R1	9,2156	0,06225	0,8951	79,4604	0,2348	2,3622	1100,0
S19	Dong	R2	2,9411	0,02394	0,9711	78,6111	0,1648	2,9411	140,0
S20	SPE	C1	14,5098	0,1089	0,8355	79,9406	0,1697	4,8387	190,0
S20	SPE	C2	6,4705	0,0263	0,8560	77,8876	0,2273	3,4883	41,5
S20	SPE	P2	3,7254	0,0047	0,8511	75,1818	0,1878	6,4516	14,5
S20	SPE	R1	1,1764	0,0	0,9919	96,3333	0,1129	0,0	7
S20	SPE	R2	2,5490	0,0167	0,8969	73,35	0,0957	12,5	7
S20	UMLN	C1	9,8039	0,0862	0,8985	72,7256	0,1209	4,9382	33,2
S20	UMLN	C2	4,5098	0,0095	0,9837	85,25	0,1739	1,9230	11,6666
S20	UMLN	P1	5,6862	0,0502	0,5516	73,4428	0,1878	0,0	12,6428
S20	UMLN	P2	3,3333	0,0071	0,7984	85,7878	0,1806	7,6923	7,6666
S20	UMLN	R2	5,8823	0,0287	0,9421	650,0	0,1119	7,1428	22

Tableau A.3 – Données oculométriques - partie « question »

A.3.2 Tableau partie « support »

Sujet	Rep	Question	DS	IA/IAN	RDF/S
S01	UMLNotation	C1	4,5098	0,9384	84,6307
S01	UMLNotation	C2	2,9411	0,0	72,4230
S01	UMLNotation	P1	9,6078	0,8622	77,0649
S01	UMLNotation	P2	12,3529	0,8631	73,9468
S01	UMLNotation	R1	2,1568	0,9879	71,4285
S01	UMLNotation	R2	6,2745	0,8717	72,75
S01	Dong	C1	5,4901	0,5873	79,4929
S01	Dong	C2	1,3725	0,9841	91,3043
S01	Dong	P1	16,6666	0,7171	79,2944
S01	Dong	R1	1,7647	0,9986	82,5522
S01	Dong	R2	2,5490	0,9945	83,0857
S02	Dong	C1	2,9411	0,9965	85,8617
S02	Dong	C2	1,1764	0,9974	70,3513
S02	Dong	P1	6,0784	0,9904	81,3904
S02	Dong	P2	8,8235	0,8352	79,0459
S02	Dong	R1	2,1568	1,0	82,4727
S02	Dong	R2	3,3333	0,9812	72,75
S02	SPE	C1	2,7450	0,9917	74,8208
S02	SPE	C2	1,1764	0,9817	88,5
S02	SPE	P1	7,6470	0,9127	84,2555
S02	SPE	R1	2,1568	1,0	96,1219
S02	SPE	R2	1,3725	1,0	87,1818
S03	SPE	C2	1,3725	0,5747	78,3
S03	SPE	P2	8,8235	0,8180	69,7011
S03	SPE	R2	2,5490	0,9809	70,0588
S03	UMLNotation	C1	3,3333	0,9526	62,04
S03	UMLNotation	C2	0,9803	1,0	64,2857
S03	UMLNotation	P1	5,8823	0,9961	75,5670
S03	UMLNotation	P2	8,8235	0,9957	71,8034
S03	UMLNotation	R1	1,3725	1,0	94,0
S03	UMLNotation	R2	2,7450	0,9503	69,0571
S04	Dong	C1	0,9803	1,0	98,4
S04	Dong	C2	0,7843	0,9886	85,7142
S04	Dong	P1	4,9019	0,9216	85,52
S04	Dong	P2	7,4509	0,9599	77,9386
S04	Dong	R1	1,5686	1,0	71,2121
S04	Dong	R2	1,3725	1,0	81,0
S04	UMLNotation	C1	1,9607	0,8319	73,5294
S04	UMLNotation	C2	0,7843	1,0	109,5714
S04	UMLNotation	P1	5,6862	0,9919	74,6302
S04	UMLNotation	P2	7,8431	0,9910	75,0730
S04	UMLNotation	R1	1,7647	1,0	78,9210
S04	UMLNotation	R2	0,9803	1,0	91,6891

Sujet	Rep	Question	DS	IA/IAN	RDF/S
S05	SPE	C1	0,3921	0,9582	66,5
S05	SPE	C2	0,5882	0,0	124,75
S05	SPE	R1	1,3725	1,0	85,1666
S05	SPE	R2	1,3725	0,9847	82,0714
S05	Dong	C1	1,9607	0,8462	75,5294
S05	Dong	C2	2,9411	0,9726	71,4
S05	Dong	R1	2,1568	1,0	89,1428
S05	Dong	R2	0,7843	1,0	64,2857
S06	UMLNotation	C1	1,1764	0,9528	73,8888
S06	UMLNotation	C2	0,5882	0,9408	64,4285
S06	UMLNotation	P2	11,9607	1,0	76,6933
S06	UMLNotation	R1	2,1568	1,0	89,7222
S06	UMLNotation	R2	2,5490	0,9617	77,9677
S06	SPE	C1	2,5490	0,9684	80,2187
S06	SPE	C2	0,7843	1,0	81,375
S06	SPE	P1	7,8431	0,8847	79,8514
S06	SPE	R1	3,5294	0,9891	73,5466
S06	SPE	R2	2,5490	0,9910	75,775
S07	UMLNotation	C1	1,1764	0,9617	118,4
S07	UMLNotation	C2	0,7843	0,9815	69,5
S07	UMLNotation	P1	3,1372	0,8853	72,1944
S07	UMLNotation	P2	5,0980	1,0	70,4166
S07	UMLNotation	R1	2,5490	0,9848	79,7916
S07	UMLNotation	R2	0,3921	1,0	90,5714
S07	Dong	C1	0,7843	0,9853	62,6
S07	Dong	C2	0,3921	0,9618	58,5
S07	Dong	P1	6,2745	0,6109	81,5535
S07	Dong	P2	7,2549	0,9317	81,0377
S07	Dong	R1	1,5686	0,9872	75,5333
S07	Dong	R2	0,9803	0,9753	83,3333
S08	Dong	C1	1,9607	0,9905	68,0416
S08	Dong	C2	1,3725	0,9944	58,25
S08	Dong	P1	7,4509	0,8962	75,5591
S08	Dong	P2	6,8627	0,5308	75,5922
S08	Dong	R1	1,1764	0,9943	85,2222
S08	Dong	R2	1,9607	0,8992	94,5555
S08	SPE	C1	2,7450	0,9751	72,5135
S08	SPE	C2	0,3921	0,9253	58,5
S08	SPE	P1	7,4509	0,9031	76,6533
S08	SPE	R1	1,1764	1,0	67,1666
S08	SPE	R2	0,5882	1,0	78,0
S09	SPE	C1	0,5882	1,0	93,5
S09	SPE	C2	0,3921	0,0	71,1818
S09	SPE	P1	4,7058	0,9721	92,36893

Sujet	Rep	Question	DS	IA/IAN	RDF/S
S09	SPE	R1	1,1764	0,9923	78,0
S09	SPE	R2	3,3333	0,9463	74,8837
S09	UMLNotation	C1	1,1764	1,0	83,4285
S09	UMLNotation	C2	0,3921	1,0	66,5
S09	UMLNotation	P1	6,4705	0,9929	89,9450
S09	UMLNotation	R2	1,1764	0,7310	55,0
S10	Dong	C1	0,3921	1,0	83,25
S10	Dong	C2	0,1960	1,0	50,0
S10	Dong	P1	4,3137	0,9604	73,7115
S10	Dong	P2	6,8627	0,9080	73,5921
S10	Dong	R1	1,9607	0,9741	79,7368
S10	Dong	R2	1,1764	0,8720	74,8
S10	UMLNotation	C2	2,5490	0,9944	86,0
S10	UMLNotation	P1	4,5098	0,9766	72,3437
S10	UMLNotation	P2	4,1176	0,9796	80,625
S10	UMLNotation	R1	0,3921	1,0	87,5
S10	UMLNotation	R2	1,7647	0,9872	65,1666
S13	UMLNotation	C1	0,1960	1,0	67,0
S13	UMLNotation	C2	0,1960	1,0	67,0
S13	UMLNotation	P2	0,7843	1,0	95,75
S13	UMLNotation	R1	0,1960	1,0	67,0
S13	Dong	C1	2,1568	0,9791	82,8148
S13	Dong	C2	1,9607	0,9975	75,3043
S13	Dong	P1	11,3725	0,6537	81,3275
S13	Dong	R1	0,9803	1,0	95,8461
S13	Dong	R2	1,3725	1,0	84,0675
S14	Dong	C1	1,3725	0,9483	66,6666
S14	Dong	C2	3,5294	0,9634	60,2222
S14	Dong	P2	12,1568	0,9297	70,9407
S14	Dong	R1	2,7450	0,9963	69,8113
S14	Dong	R2	1,1764	0,7378	63,8333
S14	SPE	C1	2,7450	0,7846	68,6875
S14	SPE	C2	1,7647	0,9043	69,4166
S14	SPE	P1	2,5490	0,7590	75,0416
S14	SPE	R1	1,5686	1,0	77,0666
S14	SPE	R2	0,9803	0,8741	81,25
S15	SPE	C1	1,1764	0,8987	72,875
S15	SPE	C2	1,9607	0,7697	65,6923
S15	SPE	P1	5,0980	0,6037	65,425
S15	SPE	P2	3,9215	0,9138	98,7717
S15	SPE	R1	2,5490	0,9813	80,1698
S15	SPE	R2	2,7450	0,9894	89,425
S15	UMLNotation	C1	1,7647	0,9362	90,1666
S15	UMLNotation	C2	0,0	1,0	1,0

Sujet	Rep	Question	DS	IA/IAN	RDF/S
S15	UMLNotation	P1	6,4705	0,9962	83,9583
S15	UMLNotation	P2	8,4313	0,9955	92,5906
S15	UMLNotation	R2	2,1568	0,9645	81,5789
S16	Dong	C2	1,5686	0,8846	96,6
S16	Dong	P1	6,6666	0,9058	76,0946
S16	UMLNotation	C1	3,3333	0,9125	71,1923
S16	UMLNotation	C2	2,3529	0,0	88,4736
S16	UMLNotation	P2	6,8627	0,8340	71,423
S18	UMLNotation	C1	0,1960	1,0	94,0
S18	UMLNotation	C2	0,3921	0,0	62,5
S18	UMLNotation	P1	3,7254	0,9296	71,6666
S18	UMLNotation	P2	4,9019	0,9724	96,912
S18	UMLNotation	R1	0,9803	1,0	72,7272
S18	SPE	C1	3,3333	0,9870	71,7142
S18	SPE	C2	0,5882	0,9730	69,3333
S18	SPE	P1	4,9019	0,9404	94,5531
S18	SPE	P2	12,1442	0,8152	100,0443
S18	SPE	R1	1,3725	0,9942	84,7142
S18	SPE	R2	0,5882	0,0	75,25
S19	UMLNotation	C1	2,5490	0,9469	67,6315
S19	UMLNotation	C2	3,1372	0,9875	78,76
S19	UMLNotation	P1	2,3529	0,9387	69,9047
S19	UMLNotation	P2	5,6862	0,9574	66,9629
S19	UMLNotation	R1	1,1764	1,0	118,0714
S19	UMLNotation	R2	8,0392	0,7094	73,5609
S19	Dong	C2	3,1372	0,9903	72,6666
S19	Dong	R1	1,9607	0,9977	73,4615
S19	Dong	R2	4,5098	0,9756	82,4102
S20	SPE	C1	2,7450	0,7801	76,0952
S20	SPE	C2	1,9607	1,0	95,75
S20	SPE	P2	5,4901	0,9403	86,5930
S20	SPE	R1	0,5882	1,0	74,8333
S20	SPE	R2	0,3921	1,0	83,3333
S20	UMLNotation	C1	2,7450	0,9484	87,6250
S20	UMLNotation	C2	0,5882	1,0	77,6666
S20	UMLNotation	P1	6,4705	0,9978	95,7634
S20	UMLNotation	P2	6,8627	0,9985	97,3631
S20	UMLNotation	R2	2,3529	1,0	95,7297

Tableau A.4 – Données oculométriques - partie « support »

A.4 Tableau des distances relatives entre chemins visuels

A.4.1 Groupe du sexe

Framework	Rep	Question	SujetA	Sujet B	DRCV
JRefactory	Dong	C1	M2	M10	263,41678502098574
JRefactory	Dong	C1	F4	F8	154,16239178412204
JRefactory	Dong	C1	F4	F14	152,04598882449926
JRefactory	Dong	C1	F8	F14	177,676649291397
JRefactory	Dong	C2	M2	M10	173,78309484307584
JRefactory	Dong	C2	M2	M16	205,11476122822899
JRefactory	Dong	C2	M10	M16	267,03528784751853
JRefactory	Dong	C2	F4	F8	335,98981811948084
JRefactory	Dong	C2	F4	F14	209,87793683480976
JRefactory	Dong	C2	F8	F14	261,44599826261674
JRefactory	Dong	P1	M2	M10	486,07879185043527
JRefactory	Dong	P1	M2	M16	596,6767410542851
JRefactory	Dong	P1	M10	M16	480,2874748033709
JRefactory	Dong	P1	F4	F8	539,4900816159856
JRefactory	Dong	P2	M2	M10	408,94900268048053
JRefactory	Dong	P2	F4	F8	362,43456933238843
JRefactory	Dong	P2	F4	F14	476,1798886471333
JRefactory	Dong	P2	F8	F14	358,7880631882915
JRefactory	Dong	R1	M2	M10	146,99329614845337
JRefactory	Dong	R1	F4	F8	216,16397242648276
JRefactory	Dong	R1	F4	F14	175,1759656561331
JRefactory	Dong	R1	F8	F14	223,93537138930463
JRefactory	Dong	R2	M2	M10	989,6110567775837
JRefactory	Dong	R2	F4	F8	253,29778918205912
JRefactory	Dong	R2	F4	F14	206,84538115473356
JRefactory	Dong	R2	F8	F14	333,91090076713294
JRefactory	SPE	C1	M9	M15	290,8309803151156
JRefactory	SPE	C1	M9	M20	540,431776980082
JRefactory	SPE	C1	M15	M20	474,3359273963681
JRefactory	SPE	C2	M3	M9	659,5177631726648
JRefactory	SPE	C2	M3	M15	510,51131059203004
JRefactory	SPE	C2	M3	M20	668,8527920359555
JRefactory	SPE	C2	M9	M15	628,3724118448325
JRefactory	SPE	C2	M9	M20	619,4833658920746
JRefactory	SPE	C2	M15	M20	441,96612064436346
JRefactory	SPE	P1	M9	M15	606,2099018203065
JRefactory	SPE	P2	M3	M15	355,24048031827596
JRefactory	SPE	P2	M3	M20	562,6750771094198
JRefactory	SPE	P2	M15	M20	437,60823142806845
JRefactory	SPE	R1	M9	M15	348,58887472485793
JRefactory	SPE	R1	M9	M20	167,64133690850971

Framework	Rep	Question	SujetA	Sujet B	DRCV
JRefactory	SPE	R1	M15	M20	221,08739712417818
JRefactory	SPE	R2	M3	M9	195,37789520544325
JRefactory	SPE	R2	M3	M15	204,38749566796636
JRefactory	SPE	R2	M3	M20	267,7557503937694
JRefactory	SPE	R2	M9	M15	230,33819937870268
JRefactory	SPE	R2	M9	M20	188,84655482460172
JRefactory	SPE	R2	M15	M20	286,9752051337159
JRefactory	UMLNotation	C1	M1	M7	472,85434009093774
JRefactory	UMLNotation	C1	M1	M18	383,81803325364666
JRefactory	UMLNotation	C1	M1	M19	408,34944244010535
JRefactory	UMLNotation	C1	M7	M18	363,90858635227414
JRefactory	UMLNotation	C1	M7	M19	865,1916761861069
JRefactory	UMLNotation	C1	M18	M19	434,11708599613274
JRefactory	UMLNotation	C1	F6	F13	303,1207045964903
JRefactory	UMLNotation	C2	M1	M7	574,4218224469747
JRefactory	UMLNotation	C2	M1	M18	546,0578562794168
JRefactory	UMLNotation	C2	M1	M19	936,2738231860195
JRefactory	UMLNotation	C2	M7	M18	460,51887967878724
JRefactory	UMLNotation	C2	M7	M19	527,348089965029
JRefactory	UMLNotation	C2	M18	M19	592,3827373848392
JRefactory	UMLNotation	C2	F6	F13	520,9476198538905
JRefactory	UMLNotation	P1	M1	M7	451,77894426137294
JRefactory	UMLNotation	P1	M1	M18	437,00321357474394
JRefactory	UMLNotation	P1	M1	M19	406,68026140405794
JRefactory	UMLNotation	P1	M7	M18	243,246301662752
JRefactory	UMLNotation	P1	M7	M19	428,60201964727963
JRefactory	UMLNotation	P1	M18	M19	507,68378168165145
JRefactory	UMLNotation	P2	M1	M7	518,7006708740414
JRefactory	UMLNotation	P2	M1	M18	663,7827821080098
JRefactory	UMLNotation	P2	M1	M19	594,2119206662769
JRefactory	UMLNotation	P2	M7	M18	491,18888829244963
JRefactory	UMLNotation	P2	M7	M19	330,6052590806655
JRefactory	UMLNotation	P2	M18	M19	534,8767703251536
JRefactory	UMLNotation	P2	F6	F13	363,3655665744816
JRefactory	UMLNotation	R1	M1	M7	531,9805717900772
JRefactory	UMLNotation	R1	M1	M18	594,5813150011622
JRefactory	UMLNotation	R1	M1	M19	432,72771129427485
JRefactory	UMLNotation	R1	M7	M18	452,99815421569525
JRefactory	UMLNotation	R1	M7	M19	492,5478170070019
JRefactory	UMLNotation	R1	M18	M19	490,2629885564156
JRefactory	UMLNotation	R1	F6	F13	404,22351699226357
JRefactory	UMLNotation	R2	M1	M7	490,7935751466142
JRefactory	UMLNotation	R2	M1	M19	414,9134257109436
JRefactory	UMLNotation	R2	M7	M19	391,8620674370928

Framework	Rep	Question	SujetA	Sujet B	DRCV
Nutch	Dong	C1	M1	M7	634,9986792018132
Nutch	Dong	C1	F5	F13	1018,9641277477107
Nutch	Dong	C2	M1	M7	457,22411534584205
Nutch	Dong	C2	M1	M19	424,5168540091684
Nutch	Dong	C2	M7	M19	232,4332511030108
Nutch	Dong	C2	F5	F13	176,89061515728744
Nutch	Dong	P1	M1	M7	618,5794410051733
Nutch	Dong	R1	M1	M7	93,90979503999037
Nutch	Dong	R1	M1	M19	248,97769380103563
Nutch	Dong	R1	M7	M19	280,80532677953465
Nutch	Dong	R1	F5	F13	178,69005238047046
Nutch	Dong	R2	M1	M7	519,4337642087075
Nutch	Dong	R2	M1	M19	244,4891484629958
Nutch	Dong	R2	M7	M19	662,7326907082237
Nutch	Dong	R2	F5	F13	412,9019347383371
Nutch	SPE	C1	M2	M18	559,172638977271
Nutch	SPE	C1	F6	F8	436,4902613431032
Nutch	SPE	C1	F6	F14	592,4537101565536
Nutch	SPE	C1	F8	F14	630,8600362050663
Nutch	SPE	C2	M2	M18	423,6106761775591
Nutch	SPE	C2	F6	F8	242,928036584778
Nutch	SPE	C2	F6	F14	186,17299021429122
Nutch	SPE	C2	F8	F14	344,629925350608
Nutch	SPE	P1	M2	M18	552,1128536117749
Nutch	SPE	P1	F6	F8	537,8305181955785
Nutch	SPE	P1	F6	F14	415,6308485237283
Nutch	SPE	P1	F8	F14	558,1711913054187
Nutch	SPE	R1	M2	M18	76,8387421511933
Nutch	SPE	R1	F6	F8	242,92718240292587
Nutch	SPE	R1	F6	F14	264,80312969845346
Nutch	SPE	R1	F8	F14	216,4085121832376
Nutch	SPE	R2	M2	M18	172,7609717562335
Nutch	SPE	R2	F6	F8	235,5210793828915
Nutch	SPE	R2	F6	F14	210,20505433439257
Nutch	SPE	R2	F8	F14	152,3471275158107
Nutch	UMLNotation	C1	M3	M9	297,0881459736255
Nutch	UMLNotation	C1	M3	M15	363,94364392630087
Nutch	UMLNotation	C1	M3	M16	391,6411113780305
Nutch	UMLNotation	C1	M3	M20	325,9749731422144
Nutch	UMLNotation	C1	M9	M15	316,3388630786579
Nutch	UMLNotation	C1	M9	M16	301,49259792402927
Nutch	UMLNotation	C1	M9	M20	505,22280779531735
Nutch	UMLNotation	C1	M15	M16	357,00857206383427
Nutch	UMLNotation	C1	M15	M20	393,80134772854615

Framework	Rep	Question	SujetA	Sujet B	DRCV
Nutch	UMLNotation	C1	M16	M20	397,97242154025776
Nutch	UMLNotation	C2	M3	M9	215,0868247743111
Nutch	UMLNotation	C2	M3	M10	288,760080747937
Nutch	UMLNotation	C2	M3	M15	322,18651608560157
Nutch	UMLNotation	C2	M3	M16	228,88762915820394
Nutch	UMLNotation	C2	M3	M20	285,75833299991046
Nutch	UMLNotation	C2	M9	M10	225,9494721687668
Nutch	UMLNotation	C2	M9	M15	310,175152204275
Nutch	UMLNotation	C2	M9	M16	278,4697096555429
Nutch	UMLNotation	C2	M9	M20	195,9235761196277
Nutch	UMLNotation	C2	M10	M15	273,46764190453285
Nutch	UMLNotation	C2	M10	M16	230,21867640609875
Nutch	UMLNotation	C2	M10	M20	299,644972672871
Nutch	UMLNotation	C2	M15	M16	210,73639909968682
Nutch	UMLNotation	C2	M15	M20	276,9603395612634
Nutch	UMLNotation	C2	M16	M20	248,06151152487027
Nutch	UMLNotation	P1	M3	M9	289,8145548678891
Nutch	UMLNotation	P1	M3	M10	478,4678877853969
Nutch	UMLNotation	P1	M3	M15	271,22124327067235
Nutch	UMLNotation	P1	M3	M20	475,90936811855136
Nutch	UMLNotation	P1	M9	M10	306,33394895881423
Nutch	UMLNotation	P1	M9	M15	264,6533145291601
Nutch	UMLNotation	P1	M9	M20	364,60144760268435
Nutch	UMLNotation	P1	M10	M15	351,71019839871184
Nutch	UMLNotation	P1	M10	M20	475,3750726603235
Nutch	UMLNotation	P1	M15	M20	345,62770150250327
Nutch	UMLNotation	P2	M3	M10	194,80901011739493
Nutch	UMLNotation	P2	M3	M15	335,50502669613144
Nutch	UMLNotation	P2	M3	M16	267,98725960876334
Nutch	UMLNotation	P2	M3	M20	207,6627797503329
Nutch	UMLNotation	P2	M10	M15	261,9146349115127
Nutch	UMLNotation	P2	M10	M16	263,55387121623
Nutch	UMLNotation	P2	M10	M20	278,28087673936915
Nutch	UMLNotation	P2	M15	M16	269,142338529877
Nutch	UMLNotation	P2	M15	M20	244,14690022265273
Nutch	UMLNotation	P2	M16	M20	237,78963456291544
Nutch	UMLNotation	R1	M3	M10	387,4651742229493
Nutch	UMLNotation	R2	M3	M9	384,63541193423384
Nutch	UMLNotation	R2	M3	M10	242,1889038875621
Nutch	UMLNotation	R2	M3	M15	249,3111759673699
Nutch	UMLNotation	R2	M3	M20	238,79495772715535
Nutch	UMLNotation	R2	M9	M10	241,34087371894503
Nutch	UMLNotation	R2	M9	M15	475,62740407246054
Nutch	UMLNotation	R2	M9	M20	397,16792445896283

Framework	Rep	Question	SujetA	Sujet B	DRCV
Nutch	UMLNotation	R2	M10	M15	215,0436955144166
Nutch	UMLNotation	R2	M10	M20	263,44300593893274
Nutch	UMLNotation	R2	M15	M20	259,7845421258684

Tableau A.5 – Données DRCV - sexe

A.4.2 Groupe du niveau d'étude

Framework	Rep	Question	SujetA	Sujet B	DRCV
JRefactory	Dong	C1	M2	M4	240,40202643117752
JRefactory	Dong	C1	M2	M10	263,41678502098574
JRefactory	Dong	C1	M4	M10	199,79585317067608
JRefactory	Dong	C1	D8	D14	177,676649291397
JRefactory	Dong	C2	M2	M4	265,1041796382023
JRefactory	Dong	C2	M2	M10	173,78309484307584
JRefactory	Dong	C2	M2	M16	205,11476122822899
JRefactory	Dong	C2	M4	M10	188,4948061877666
JRefactory	Dong	C2	M4	M16	230,14921206437523
JRefactory	Dong	C2	M10	M16	267,03528784751853
JRefactory	Dong	C2	D8	D14	261,44599826261674
JRefactory	Dong	P1	M2	M4	619,3480855827116
JRefactory	Dong	P1	M2	M10	486,07879185043527
JRefactory	Dong	P1	M2	M16	596,6767410542851
JRefactory	Dong	P1	M4	M10	601,6888286086858
JRefactory	Dong	P1	M4	M16	494,80999755009566
JRefactory	Dong	P1	M10	M16	480,2874748033709
JRefactory	Dong	P2	M2	M4	418,63094096806896
JRefactory	Dong	P2	M2	M10	408,94900268048053
JRefactory	Dong	P2	M4	M10	518,5873672085099
JRefactory	Dong	P2	D8	D14	358,7880631882915
JRefactory	Dong	R1	M2	M4	173,12495764022847
JRefactory	Dong	R1	M2	M10	146,99329614845337
JRefactory	Dong	R1	M4	M10	150,33112270113364
JRefactory	Dong	R1	D8	D14	223,93537138930463
JRefactory	Dong	R2	M2	M4	166,08473896316903
JRefactory	Dong	R2	M2	M10	989,6110567775837
JRefactory	Dong	R2	M4	M10	627,1828823968018
JRefactory	Dong	R2	D8	D14	333,91090076713294
JRefactory	SPE	C1	M9	M20	540,431776980082
JRefactory	SPE	C1	D5	D15	306,01477601064397
JRefactory	SPE	C2	M3	M9	659,5177631726648
JRefactory	SPE	C2	M3	M20	668,8527920359555
JRefactory	SPE	C2	M9	M20	619,4833658920746

Framework	Rep	Question	SujetA	Sujet B	DRCV
JRefactory	SPE	C2	D5	D15	397,7368702790442
JRefactory	SPE	P2	M3	M20	562,6750771094198
JRefactory	SPE	R1	M9	M20	167,64133690850971
JRefactory	SPE	R1	D5	D15	165,35997961249805
JRefactory	SPE	R2	M3	M9	195,37789520544325
JRefactory	SPE	R2	M3	M20	267,7557503937694
JRefactory	SPE	R2	M9	M20	188,84655482460172
JRefactory	SPE	R2	D5	D15	221,11051768258997
JRefactory	UMLNotation	C1	D1	D6	363,59858730525923
JRefactory	UMLNotation	C1	D1	D13	346,09521028545265
JRefactory	UMLNotation	C1	D1	D18	383,81803325364666
JRefactory	UMLNotation	C1	D1	D19	408,34944244010535
JRefactory	UMLNotation	C1	D6	D13	303,1207045964903
JRefactory	UMLNotation	C1	D6	D18	453,47429389049057
JRefactory	UMLNotation	C1	D6	D19	295,75000846177346
JRefactory	UMLNotation	C1	D13	D18	213,9883330023601
JRefactory	UMLNotation	C1	D13	D19	292,26489568838844
JRefactory	UMLNotation	C1	D18	D19	434,11708599613274
JRefactory	UMLNotation	C2	D1	D6	607,1133066310733
JRefactory	UMLNotation	C2	D1	D13	389,8055027968066
JRefactory	UMLNotation	C2	D1	D18	546,0578562794168
JRefactory	UMLNotation	C2	D1	D19	936,2738231860195
JRefactory	UMLNotation	C2	D6	D13	520,9476198538905
JRefactory	UMLNotation	C2	D6	D18	374,7663344764155
JRefactory	UMLNotation	C2	D6	D19	499,17887852267233
JRefactory	UMLNotation	C2	D13	D18	311,0512311112266
JRefactory	UMLNotation	C2	D13	D19	702,1852141474249
JRefactory	UMLNotation	C2	D18	D19	592,3827373848392
JRefactory	UMLNotation	P1	D1	D18	437,00321357474394
JRefactory	UMLNotation	P1	D1	D19	406,68026140405794
JRefactory	UMLNotation	P1	D18	D19	507,68378168165145
JRefactory	UMLNotation	P2	D1	D6	337,3217606257463
JRefactory	UMLNotation	P2	D1	D13	454,69003324831755
JRefactory	UMLNotation	P2	D1	D18	663,7827821080098
JRefactory	UMLNotation	P2	D1	D19	594,2119206662769
JRefactory	UMLNotation	P2	D6	D13	363,3655665744816
JRefactory	UMLNotation	P2	D6	D18	431,22292234172903
JRefactory	UMLNotation	P2	D6	D19	361,5089707727894
JRefactory	UMLNotation	P2	D13	D18	678,704325846941
JRefactory	UMLNotation	P2	D13	D19	551,825518171943
JRefactory	UMLNotation	P2	D18	D19	534,8767703251536
JRefactory	UMLNotation	R1	D1	D6	419,5255909874109
JRefactory	UMLNotation	R1	D1	D13	337,60699488087056
JRefactory	UMLNotation	R1	D1	D18	594,5813150011622

Framework	Rep	Question	SujetA	Sujet B	DRCV
JRefactory	UMLNotation	R1	D1	D19	432,72771129427485
JRefactory	UMLNotation	R1	D6	D13	404,22351699226357
JRefactory	UMLNotation	R1	D6	D18	478,7813699222711
JRefactory	UMLNotation	R1	D6	D19	348,4127114036561
JRefactory	UMLNotation	R1	D13	D18	346,6007123528669
JRefactory	UMLNotation	R1	D13	D19	331,9431754767405
JRefactory	UMLNotation	R1	D18	D19	490,2629885564156
JRefactory	UMLNotation	R2	D1	D6	867,1307955482943
JRefactory	UMLNotation	R2	D1	D19	414,9134257109436
JRefactory	UMLNotation	R2	D6	D19	249,0583153771332
Nutch	Dong	C1	D1	D5	676,6576701777363
Nutch	Dong	C1	D1	D13	834,4215923851455
Nutch	Dong	C1	D5	D13	1018,9641277477107
Nutch	Dong	C2	D1	D5	380,5461192779946
Nutch	Dong	C2	D1	D13	428,28902771035774
Nutch	Dong	C2	D1	D19	424,5168540091684
Nutch	Dong	C2	D5	D13	176,89061515728744
Nutch	Dong	C2	D5	D19	148,39328808405259
Nutch	Dong	C2	D13	D19	242,1739697587981
Nutch	Dong	P1	D1	D13	487,2015562940107
Nutch	Dong	R1	D1	D5	114,3682824785381
Nutch	Dong	R1	D1	D13	161,52726646529158
Nutch	Dong	R1	D1	D19	248,97769380103563
Nutch	Dong	R1	D5	D13	178,69005238047046
Nutch	Dong	R1	D5	D19	309,5866706553852
Nutch	Dong	R1	D13	D19	364,262429159672
Nutch	Dong	R2	D1	D5	269,53014528715414
Nutch	Dong	R2	D1	D13	376,35316102132055
Nutch	Dong	R2	D1	D19	244,4891484629958
Nutch	Dong	R2	D5	D13	412,9019347383371
Nutch	Dong	R2	D5	D19	256,94084728529845
Nutch	Dong	R2	D13	D19	343,7837871104675
Nutch	SPE	C1	D6	D8	436,4902613431032
Nutch	SPE	C1	D6	D14	592,4537101565536
Nutch	SPE	C1	D6	D18	881,2406756793154
Nutch	SPE	C1	D8	D14	630,8600362050663
Nutch	SPE	C1	D8	D18	608,7209981885335
Nutch	SPE	C1	D14	D18	755,3328468542069
Nutch	SPE	C2	D6	D8	242,928036584778
Nutch	SPE	C2	D6	D14	186,17299021429122
Nutch	SPE	C2	D6	D18	370,8562590671091
Nutch	SPE	C2	D8	D14	344,629925350608
Nutch	SPE	C2	D8	D18	784,810961756898
Nutch	SPE	C2	D14	D18	482,25143944047056

Framework	Rep	Question	SujetA	Sujet B	DRCV
Nutch	SPE	P1	D6	D8	537,8305181955785
Nutch	SPE	P1	D6	D14	415,6308485237283
Nutch	SPE	P1	D6	D18	434,19066186260295
Nutch	SPE	P1	D8	D14	558,1711913054187
Nutch	SPE	P1	D8	D18	458,7356579521314
Nutch	SPE	P1	D14	D18	364,02707656360735
Nutch	SPE	R1	D6	D8	242,92718240292587
Nutch	SPE	R1	D6	D14	264,80312969845346
Nutch	SPE	R1	D6	D18	244,36373657511825
Nutch	SPE	R1	D8	D14	216,4085121832376
Nutch	SPE	R1	D8	D18	102,36386452326403
Nutch	SPE	R1	D14	D18	187,01607384074174
Nutch	SPE	R2	D6	D8	235,5210793828915
Nutch	SPE	R2	D6	D14	210,20505433439257
Nutch	SPE	R2	D6	D18	244,86607129897666
Nutch	SPE	R2	D8	D14	152,3471275158107
Nutch	SPE	R2	D8	D18	233,88215330152735
Nutch	SPE	R2	D14	D18	199,24798738328363
Nutch	UMLNotation	C1	M3	M4	286,8438022141589
Nutch	UMLNotation	C1	M3	M9	297,0881459736255
Nutch	UMLNotation	C1	M3	M16	391,6411113780305
Nutch	UMLNotation	C1	M3	M20	325,9749731422144
Nutch	UMLNotation	C1	M4	M9	294,65041274342815
Nutch	UMLNotation	C1	M4	M16	298,3666227534357
Nutch	UMLNotation	C1	M4	M20	352,112927503824
Nutch	UMLNotation	C1	M9	M16	301,49259792402927
Nutch	UMLNotation	C1	M9	M20	505,22280779531735
Nutch	UMLNotation	C1	M16	M20	397,97242154025776
Nutch	UMLNotation	C2	M3	M4	337,3583955296986
Nutch	UMLNotation	C2	M3	M9	215,0868247743111
Nutch	UMLNotation	C2	M3	M10	288,760080747937
Nutch	UMLNotation	C2	M3	M16	228,88762915820394
Nutch	UMLNotation	C2	M3	M20	285,75833299991046
Nutch	UMLNotation	C2	M4	M9	322,24599037824544
Nutch	UMLNotation	C2	M4	M10	347,2118575889325
Nutch	UMLNotation	C2	M4	M16	339,1888359327744
Nutch	UMLNotation	C2	M4	M20	434,44488382548263
Nutch	UMLNotation	C2	M9	M10	225,9494721687668
Nutch	UMLNotation	C2	M9	M16	278,4697096555429
Nutch	UMLNotation	C2	M9	M20	195,9235761196277
Nutch	UMLNotation	C2	M10	M16	230,21867640609875
Nutch	UMLNotation	C2	M10	M20	299,644972672871
Nutch	UMLNotation	C2	M16	M20	248,06151152487027
Nutch	UMLNotation	P1	M3	M4	274,00834030123065

Framework	Rep	Question	SujetA	Sujet B	DRCV
Nutch	UMLNotation	P1	M3	M9	289,8145548678891
Nutch	UMLNotation	P1	M3	M10	478,4678877853969
Nutch	UMLNotation	P1	M3	M20	475,90936811855136
Nutch	UMLNotation	P1	M4	M9	212,7162639658616
Nutch	UMLNotation	P1	M4	M10	231,10142887500297
Nutch	UMLNotation	P1	M4	M20	398,50448877900436
Nutch	UMLNotation	P1	M9	M10	306,33394895881423
Nutch	UMLNotation	P1	M9	M20	364,60144760268435
Nutch	UMLNotation	P1	M10	M20	475,3750726603235
Nutch	UMLNotation	P2	M3	M4	235,49271664916847
Nutch	UMLNotation	P2	M3	M10	194,80901011739493
Nutch	UMLNotation	P2	M3	M16	267,98725960876334
Nutch	UMLNotation	P2	M3	M20	207,6627797503329
Nutch	UMLNotation	P2	M4	M10	267,31248941348736
Nutch	UMLNotation	P2	M4	M16	157,53211747540894
Nutch	UMLNotation	P2	M4	M20	271,0296400022052
Nutch	UMLNotation	P2	M10	M16	263,55387121623
Nutch	UMLNotation	P2	M10	M20	278,28087673936915
Nutch	UMLNotation	P2	M16	M20	237,78963456291544
Nutch	UMLNotation	R1	M3	M4	271,8037025138
Nutch	UMLNotation	R1	M3	M10	387,4651742229493
Nutch	UMLNotation	R1	M4	M10	313,0932178228112
Nutch	UMLNotation	R2	M3	M4	166,83034067699165
Nutch	UMLNotation	R2	M3	M9	384,63541193423384
Nutch	UMLNotation	R2	M3	M10	242,1889038875621
Nutch	UMLNotation	R2	M3	M20	238,79495772715535
Nutch	UMLNotation	R2	M4	M9	451,12809145640205
Nutch	UMLNotation	R2	M4	M10	161,79798034046587
Nutch	UMLNotation	R2	M4	M20	170,52982203177575
Nutch	UMLNotation	R2	M9	M10	241,34087371894503
Nutch	UMLNotation	R2	M9	M20	397,16792445896283
Nutch	UMLNotation	R2	M10	M20	263,44300593893274

Tableau A.6 – Données DRCV - niveau d'étude

A.5 Tableau de la mesure subjective

Subject	Framework	Rep	Charge mentale NASA-TLX
S01	UMLN	JRefactory	125
S01	Dong	Nutch	91
S02	Dong	JRefactory	40
S02	SPE	Nutch	40
S03	SPE	JRefactory	75
S03	UMLN	Nutch	81
S04	Dong	JRefactory	49
S04	UMLN	Nutch	62
S05	SPE	JRefactory	114
S05	Dong	Nutch	110
S06	UMLN	JRefactory	50
S06	SPE	Nutch	49
S07	UMLN	JRefactory	66
S07	Dong	Nutch	89
S08	Dong	JRefactory	48
S08	SPE	Nutch	42
S09	SPE	JRefactory	75
S09	UMLN	Nutch	81
S10	Dong	JRefactory	67
S10	UMLN	Nutch	46
S13	UMLN	JRefactory	67
S13	Dong	Nutch	158
S14	Dong	JRefactory	92
S14	SPE	Nutch	104
S15	SPE	JRefactory	50
S15	UMLN	Nutch	107
S16	Dong	JRefactory	89
S16	UMLN	Nutch	72
S18	UMLN	JRefactory	54
S18	SPE	Nutch	69
S19	UMLN	JRefactory	74
S19	Dong	Nutch	185
S20	SPE	JRefactory	54
S20	UMLN	Nutch	59

Tableau A.7 – Données NASA-TLX

Questionnaire de l'expérience



Experiment - Sujet X

questionnaire d'expérimentation

Maître de stage: GUÉHÉNEUC Yann-Gaël

Ptidej Lab
2010-2011

CHRISTOPHE Julien & SOYEUR Nicolas

August 26, 2012

1 Procédure expérimentale

Pendant l'expérience, vous allez répondre à une série de questions se rapportant à la compréhension de design patterns représentés dans un diagramme de classe UML. Vous allez devoir analyser deux diagrammes différents et répondre à six questions sur chaque. Les questions concernent les tâches suivantes :

1. Tâche de composition de pattern.
2. Tâche de participation de classes.
3. Tâche d'identification de rôles joués par une classe.

Cette expérience nécessite plus ou moins 1 heure. S'il vous plaît ne lisez/répondez pas à une question avant une autre.

Les mouvements de vos yeux vont être enregistrés par un eye-tracker. **Pour éviter des problèmes d'enregistrement durant l'expérience, évitez, le plus possible, les mouvements de la tête ou des mouvements qui pourraient nuire à l'enregistrement de votre regard (boire quelque chose, mettre quelque chose devant votre visage, etc...).** Les données collectées durant cet expérience sont totalement **anonymes**. Vous pouvez quitter l'expérience à tout moment, pour n'importe quelle raison sans pénalité d'aucune sorte. Si vous avez n'importe quelle question, n'hésitez pas à nous appeler.

L'expérience va suivre les étapes suivantes :

1. Nous allons calibrer l'eye-tracker et nous allons vous montrer **trois images de base**. Sur la première image, vous devez suivre avec vos yeux les lignes présentes sur l'image. Les deuxième et troisième images représentent des points placés à différents endroits de l'écran. Vous devez regarder 2 à 3 secondes sur chaque point de cette image. Cette étape est utilisée pour s'assurer que l'eye-tracker est calibré correctement. Après chacune de ces trois images, vous devez appuyer sur la **BARRE D'ESPACE** pour accéder à la suivante.
2. Lisez la première question, et appuyez sur la **BARRE D'ESPACE** lorsque vous êtes prêt à commencer. Il est important que vous ayez compris la question, appelez-nous si vous avez un problème.
3. Une image blanche avec une croix au milieu va apparaître à l'écran, s'il vous plaît fixez cette croix, et appuyez sur la **BARRE D'ESPACE** quand vous êtes prêt à réaliser la première question.
4. Un diagramme va s'afficher sur l'écran. La question est visible dans le coin supérieur gauche du diagramme. Vous devez penser à la réponse sans poser votre regard hors de l'écran.
5. Lorsque vous avez la réponse, appuyez sur la **BARRE D'ESPACE**. Un écran blanc va apparaître, appuyez à nouveau sur la **BARRE D'ESPACE**. Vous pouvez maintenant écrire votre réponse. Le diagramme est toujours présent, utilisez-le uniquement à titre de support. **Pendant l'écriture de la réponse, votre regard est toujours enregistré, essayez donc d'éviter de faire des mouvements de la tête ou autres mouvements qui pourraient nuire à l'enregistrement de votre regard.**
6. Lorsque vous avez répondu à la question, vous pouvez lire la suivante et appuyez sur la **BARRE D'ESPACE** et une image blanche avec une croix au milieu va à nouveau apparaître. Appuyez sur la **BARRE D'ESPACE** lorsque vous êtes prêt à réaliser la question.
7. Une fois que vous avez répondu à toutes les questions d'un diagramme, vous devrez répondre à une courte évaluation subjective.
8. Les étapes sont ensuite reproduites pour le second diagramme avec une autre représentation.

2 Bon à savoir

Dans cette section, nous allons vous donner quelques points importants à savoir avant le début de l'expérience :

- Pour plus de clarté, la plupart des attributs et des méthodes des classes ont été supprimés des diagrammes. Seuls les attributs et les opérations intéressants pour les design patterns représentés ont été gardés.
- Si la question concerne toutes les classes qui jouent un rôle dans un design pattern, alors les interfaces doivent être prises en compte.
- Si une question demande de citer plusieurs classes, les interfaces doivent être prises en considération aussi.
- Si la réponse à une question est un ou plusieurs rôles, s'il vous plaît n'écrivez que le nom des rôles et **pas le design pattern qui est associé avec**.
- Durant l'expérience, évitez le plus possible tout mouvement de la tête.

3 Diagram 1 - JRefactory

3.1 Context

JRefactory est un outil de refactoring pour le langage de programmation Java. Il inclut l'imprimante JavaStyle, une visionneuse de diagrammes de classes UML, un vérificateur de standards du code et il calcule des métriques sur le programme. Il peut coopérer avec quelques IDE (NetBeans, JBuilder, ...) sous forme de plugin mais il peut aussi être utilisé en ligne de commande.

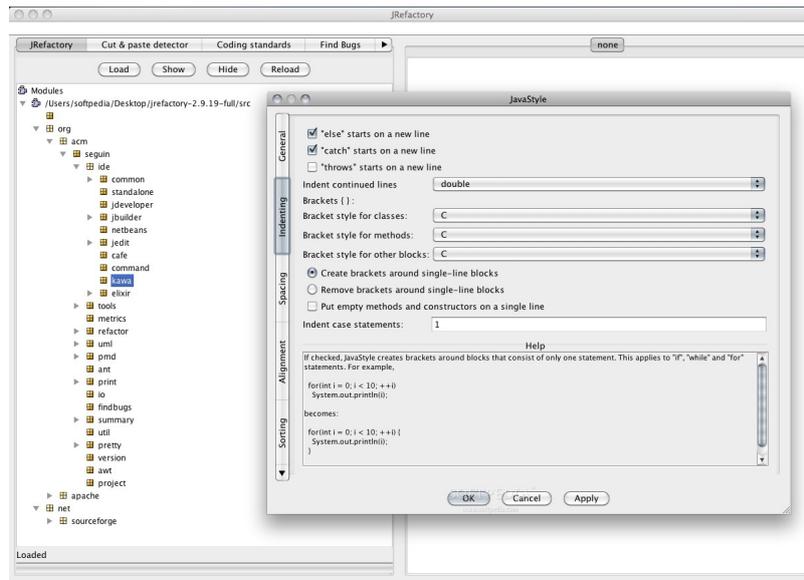


Figure 1: L'interface de JRefactory

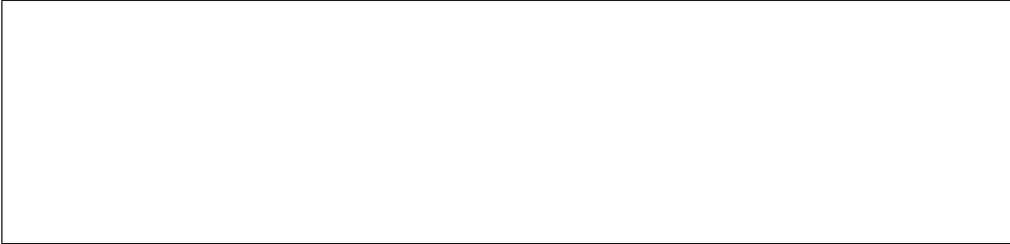
3.2 Tâche de composition de Pattern

1.) Identifiez tous les design patterns dans lesquels participe la classe UMLNode.

Réponse :

2.) Identifiez tous les design patterns dans lesquels participe la classe AdMethodTypeVisitor.

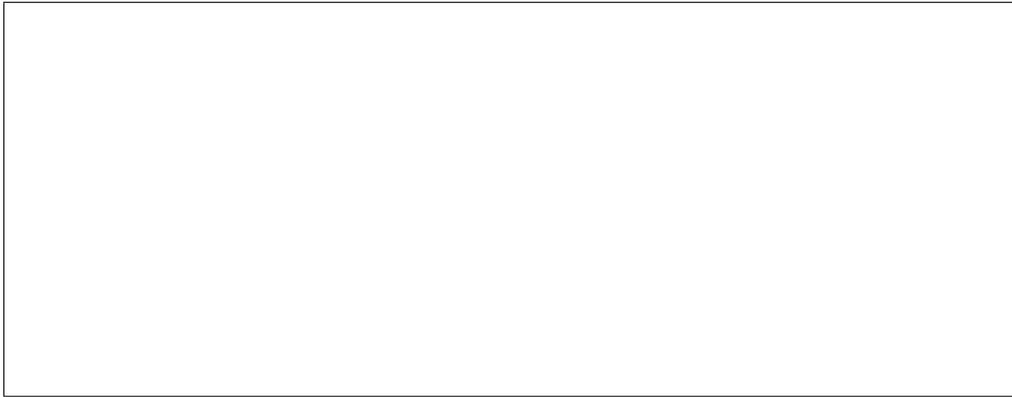
Réponse :



3.3 Tâche de participation de classes

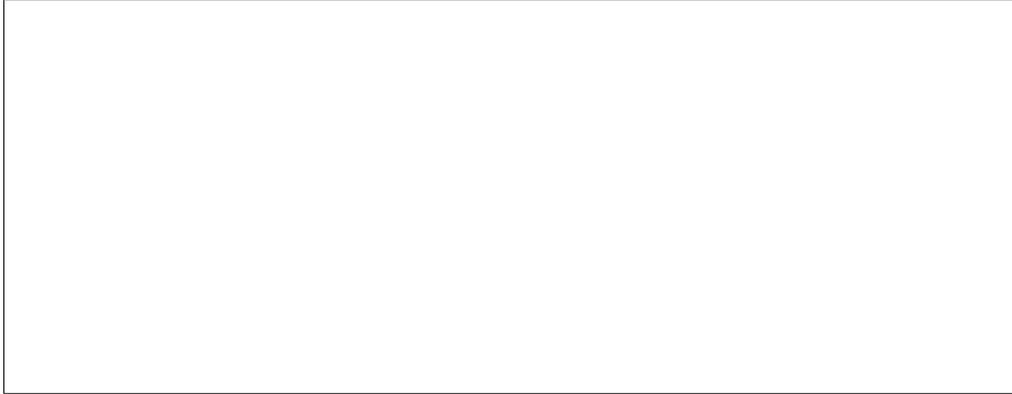
3.) Identifiez toutes les classes qui participent aux deux instances du design pattern Builder.

Réponse :



4.) Identifiez toutes les classes qui participent au design pattern Visitor.

Réponse :



3.4 Tâche des rôles joués par une classe

5.) Identifiez tous les rôles que joue la classe FileSummary.

Réponse :

6.) Identifiez tous les rôles que joue la classe UMLNodeViewerFactory.

Réponse :

3.5 Évaluation

Cette section est utilisée pour évaluer votre effort concernant les questions sur JRefactory.

Demande Mentale Quel niveau de demande mentale et d'activité perceptive étaient nécessaire (par exemple, penser, décider, calcul, mémoire, regarder, chercher, etc)? Les tâches étaient-elles faciles ou exigeantes, simples ou complexes?

Faible Moyen Haut

Demande physique Quel niveau d'activité physique était nécessaire par exemple pousser, tirer, tourner, etc? Le tâche physique était-elle facile ou exigeante, lente ou rapide, reposante ou laborieuse?

Faible Moyen Haut

Demande temporelle Comment était le niveau de pression temporelle que vous avez ressenti en raison de la vitesse ou l'allure à laquelle les éléments des tâches se sont produits?

Faible Moyen Haut

Effort Était-ce difficile (mentalement et physiquement) d'accomplir votre niveau de performance?

Faible Moyen Haut

Performance Comment pensez-vous avoir réussi l'accomplissement des objectifs de la tâche fixée par l'analyste (ou vous-même)? A quel niveau êtes-vous satisfait de votre performance dans l'accomplissement de ces objectifs?

Parfait Moyenne Échec

Niveau de frustration Comment était votre niveau d'insécurité, de découragement, d'irritation, de stress et de contrariété lors de la tâche?

Faible Moyen Haut

4 Diagram 2 - Nutch

4.1 Context

Apache Nutch est un projet open source de recherche web écrit en Java. Il permet d'indexer des sites Web, intranets et des systèmes de fichier. Il supporte de nombreux formats de fichiers différents (HTML, PDF, MS Office, OpenOffice, ...) et permet l'analyse de plus de 50 langues.

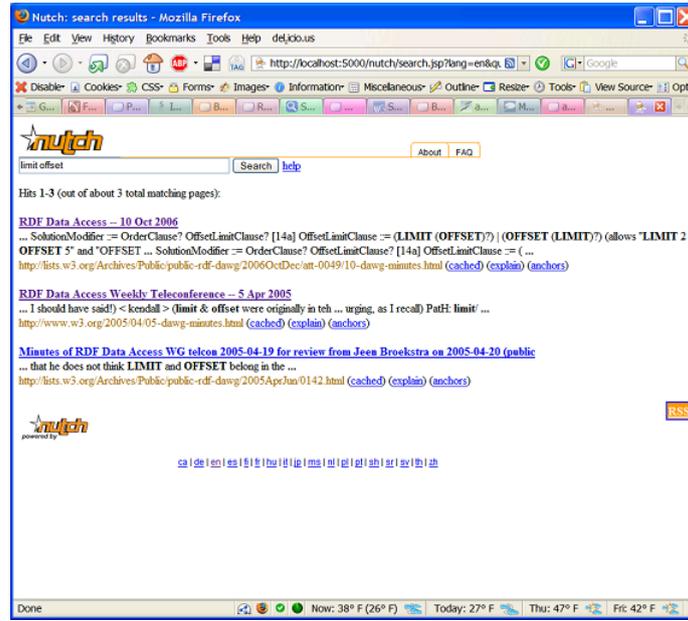


Figure 2: Une démo d'utilisation de Nutch

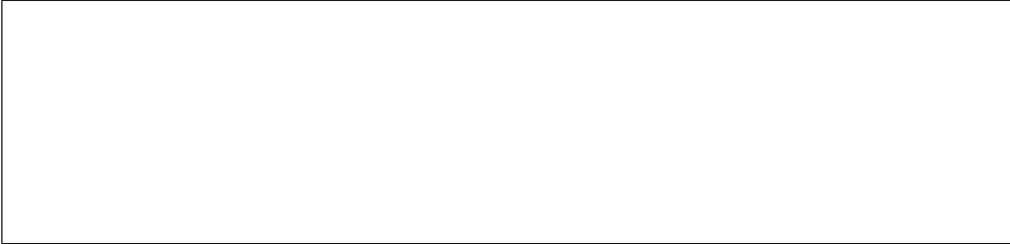
4.2 Tâche de composition de Pattern

1.) Identifiez tous les design patterns dans lesquels participe la classe Page.

Réponse :

2.) Identifiez tous les design patterns dans lesquels participe la classe FetchList-Tool.

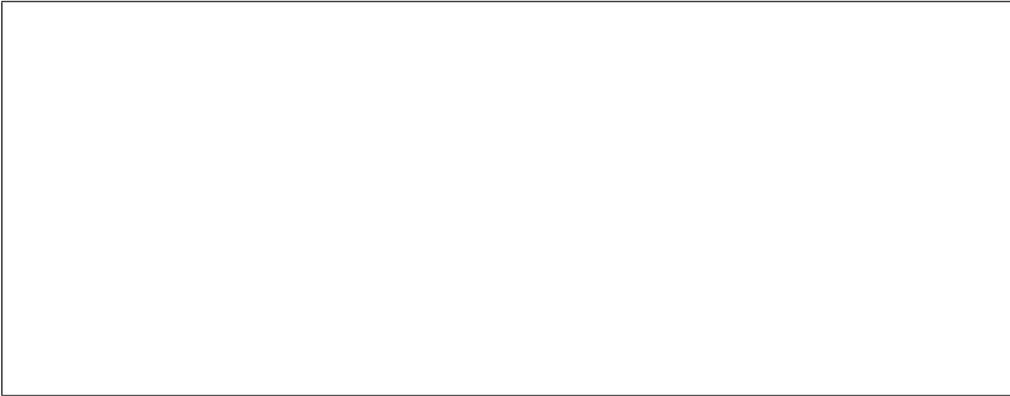
Réponse :



4.3 Tâche de participation de classes

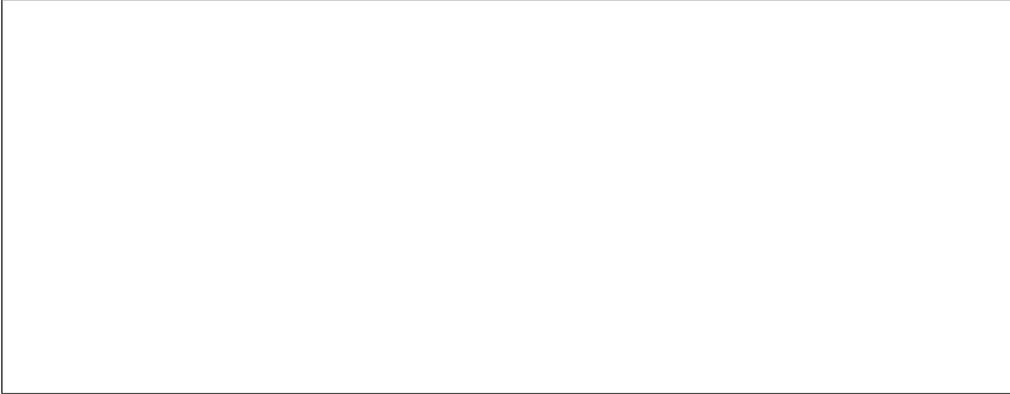
3.) Identifiez toutes les classes qui participent au design pattern Command.

Réponse :



4.) Identifiez toutes les classes qui participent au design pattern Iterator.

Réponse :



4.4 Tâche des rôles joués par une classe

5.) Identifiez tous les rôles que joue la classe `WebDBWriter`.

Réponse :

6.) Identifiez tous les rôles que joue la classe `PagesByUrlProcessor`.

Réponse :

4.5 Évaluation

Cette section est utilisée pour évaluer votre effort concernant les questions sur JRefactory.

Demande Mentale Quel niveau de demande mentale et d'activité perceptive étaient nécessaire (par exemple, penser, décider, calcul, mémoire, regarder, chercher, etc)? Les tâches étaient-elles faciles ou exigeantes, simples ou complexes?

Faible Moyan Haut

Demande physique Quel niveau d'activité physique était nécessaire par exemple pousser, tirer, tourner, etc? Le tâche physique était-elle facile ou exigeante, lente ou rapide, reposante ou laborieuse?

Faible Moyan Haut

Demande temporelle Comment était le niveau de pression temporelle que vous avez ressenti en raison de la vitesse ou l'allure à laquelle les éléments des tâches se sont produits?

Faible Moyan Haut

Effort Était-ce difficile (mentalement et physiquement) d'accomplir votre niveau de performance?

Faible Moyan Haut

Performance Comment pensez-vous avoir réussi l'accomplissement des objectifs de la tâche fixée par l'analyste (ou vous-même)? A quel niveau êtes-vous satisfait de votre performance dans l'accomplissement de ces objectifs?

Parfait Moyenne Échec

Niveau de frustration Comment était votre niveau d'insécurité, de découragement, d'irritation, de stress et de contrariété lors de la tâche?

Faible Moyan Haut

Questionnaire post-expérimental

1 Questionnaire Post-expérimental

1.1 Divers

1. Numéro du sujet ?
2. Vous êtes...
 - Homme
 - Femme
3. Quel est votre plus grand grade ?
 - Bac
 - Master
 - Docteur
 - Autre

1.2 Design Pattern

1. Qu'est qu'un Design Pattern et quelle est son utilité ?
 - Les Design Patterns vous aident à optimiser votre code source afin de faire un programme qui utilise moins de ressources et qui a moins de bug que les autres programmes.
 - Les Design Patterns vous aident à choisir des alternatives de conception qui rend le système réutilisable et évite les alternatives qui compromettent sa réutilisabilité.
 - Les Design Patterns vous aident à utiliser de meilleures annotations pour vos classes/attributs/méthodes dans votre projet.
 - Les Design Patterns vous aident à organiser le diagramme de classe avec efficacité.
 - Ne sais pas.
2. Une classe ne peut participer qu'à un seul Design Pattern.
 - Vrai
 - Faux
 - Ne sais pas
3. Une classe peut jouer différents rôles dans le même Design Pattern ?
 - Vrai
 - Faux
 - Ne sais pas
4. Quel est le but du Design Pattern "Singleton" ?

5. Quel est le Design Pattern qui correspond à cette structure ? (représentée par la Figure 1)
 - Abstraction
 - Bridge
 - Implementor
 - Iterator
 - Visitor
 - Ne sais pas

6. Représente la structure du Design Pattern "Strategy"

1.3 UML

1. Combien d'années d'expérience en UML avez-vous ?

2. Dans un diagramme de classe UML, un attribut "protected" est précédé par... ?
 - +
 - -
 - #
 - Ne sais pas
3. Dans un diagramme de classe UML, une relation de généralisation/ spécialisation est représentée par... ?
 - Un carré
 - Un losange
 - Un triangle
 - Un cercle
 - Rien
 - Ne sais pas
4. Quelle cardinalité est utilisée pour représenter une relation "au moins 1" ?

5. Quel est le nom de cette relation ? (représentée par la Figure 2)
 - Composition
 - Aggrégation
 - Réalisation / Implémentation
 - Généralisation
 - Autre
 - Ne sais pas
6. Un attribut (sans tenir compte qu'il pourrait être static) c'est...
 - Une donnée ayant une valeur pour chaque objet.
 - Une donnée ayant une valeur pour chaque classe.
 - Un domaine de valeurs.

- Ne sais pas.
- 7. Quel format est utilisé pour définir une classe abstraite ?
 - Gras
 - Italique
 - souligné
 - Ne sais pas
- 8. Si une relation de spécialisation existe entre deux classes, alors...
 - La sous-classe hérite des attributs du parent.
 - La sous-classe hérite des attributs des autres sous-classes du parent.
 - La sous-classe doit redéfinir toutes les méthodes du parent.
 - Ne sais pas.

1.4 Expérience personnelle

1. Avez-vous déjà géré (ajouté, supprimé, modifié ou maintenu) un diagramme de classe UML de plus de 20 classes ?

1.5 Commentaires

1. Avez-vous une suggestion/critique par rapport à l'expérience ?

▼ Structure

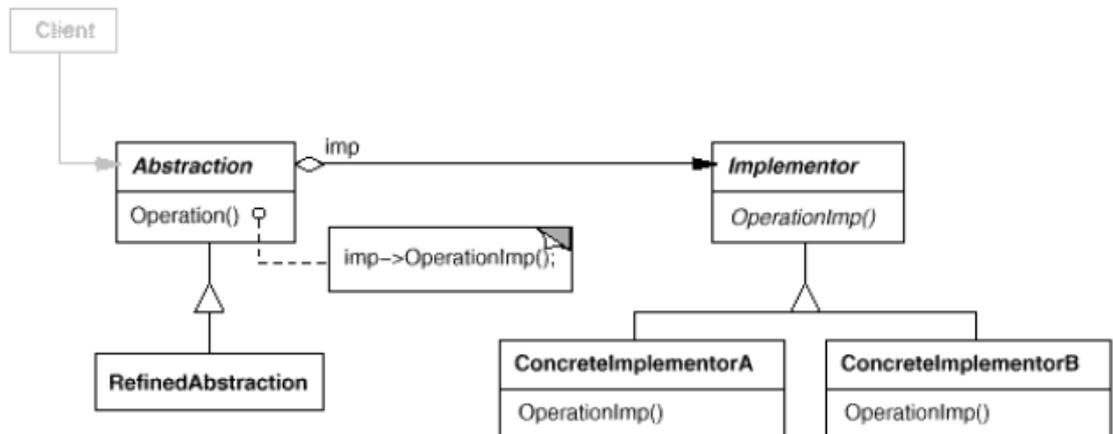


FIGURE 1 – Design Pattern

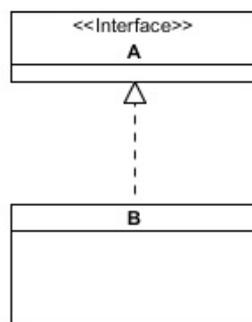


FIGURE 2 – Relation en UML

Diagrammes de l'expérience

D.1 *Nutch / Dong*

D.2 Nutch / SPE

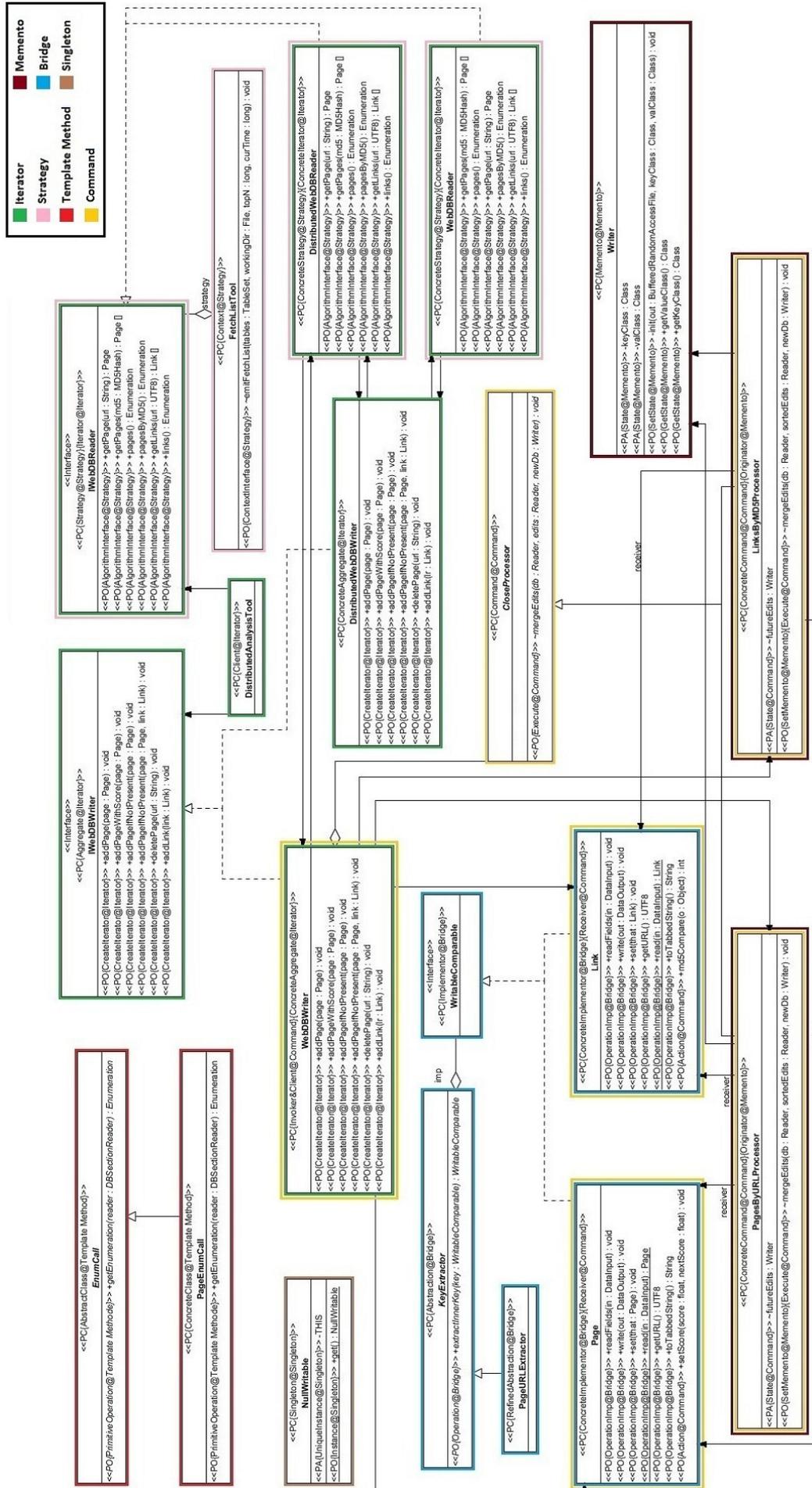


FIGURE D.2 – Nutch / SPE

D.3 Nutch / UMLNotation

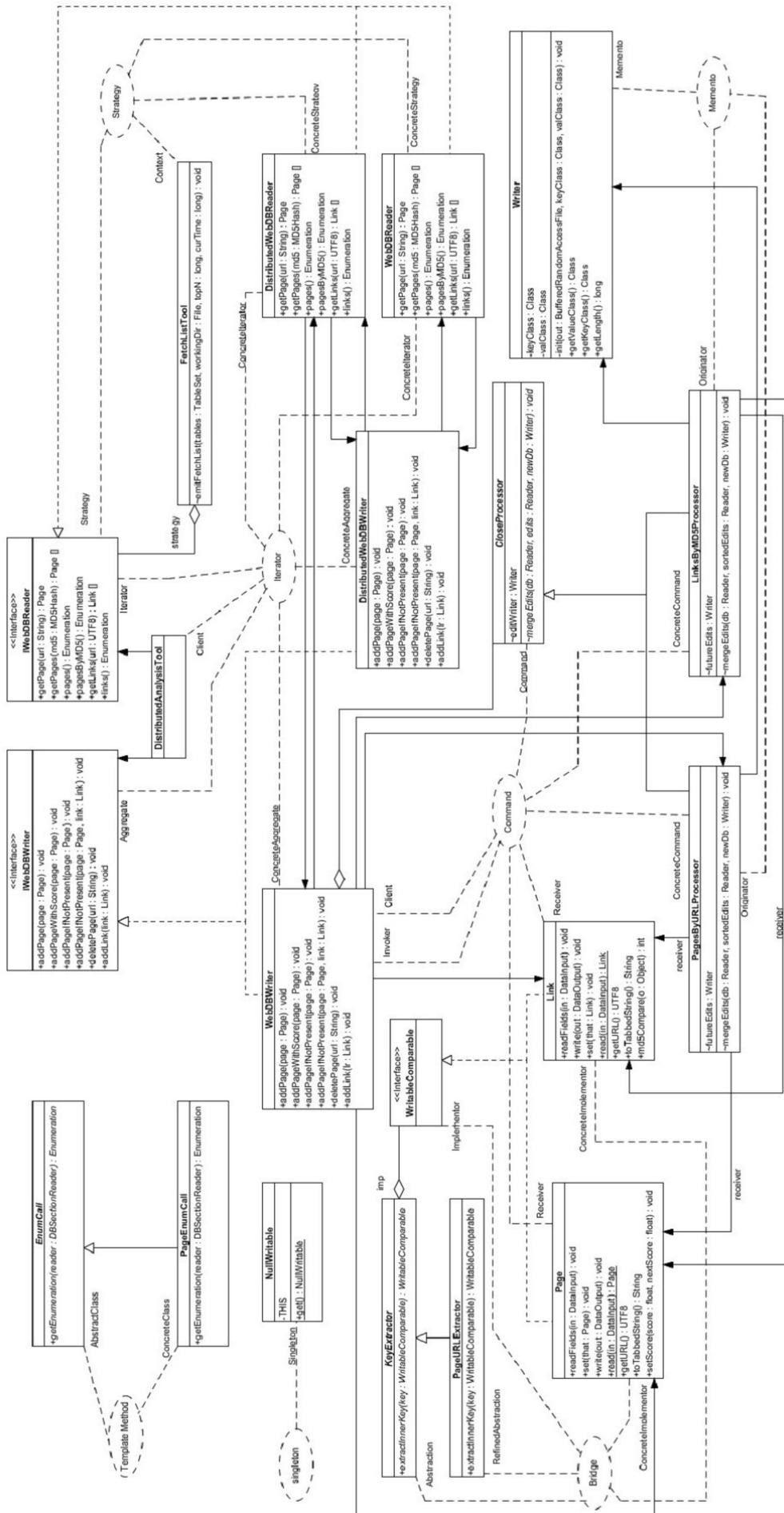


FIGURE D.3 – Nutch / UMLNotation

D.4 JRefactory / Dong

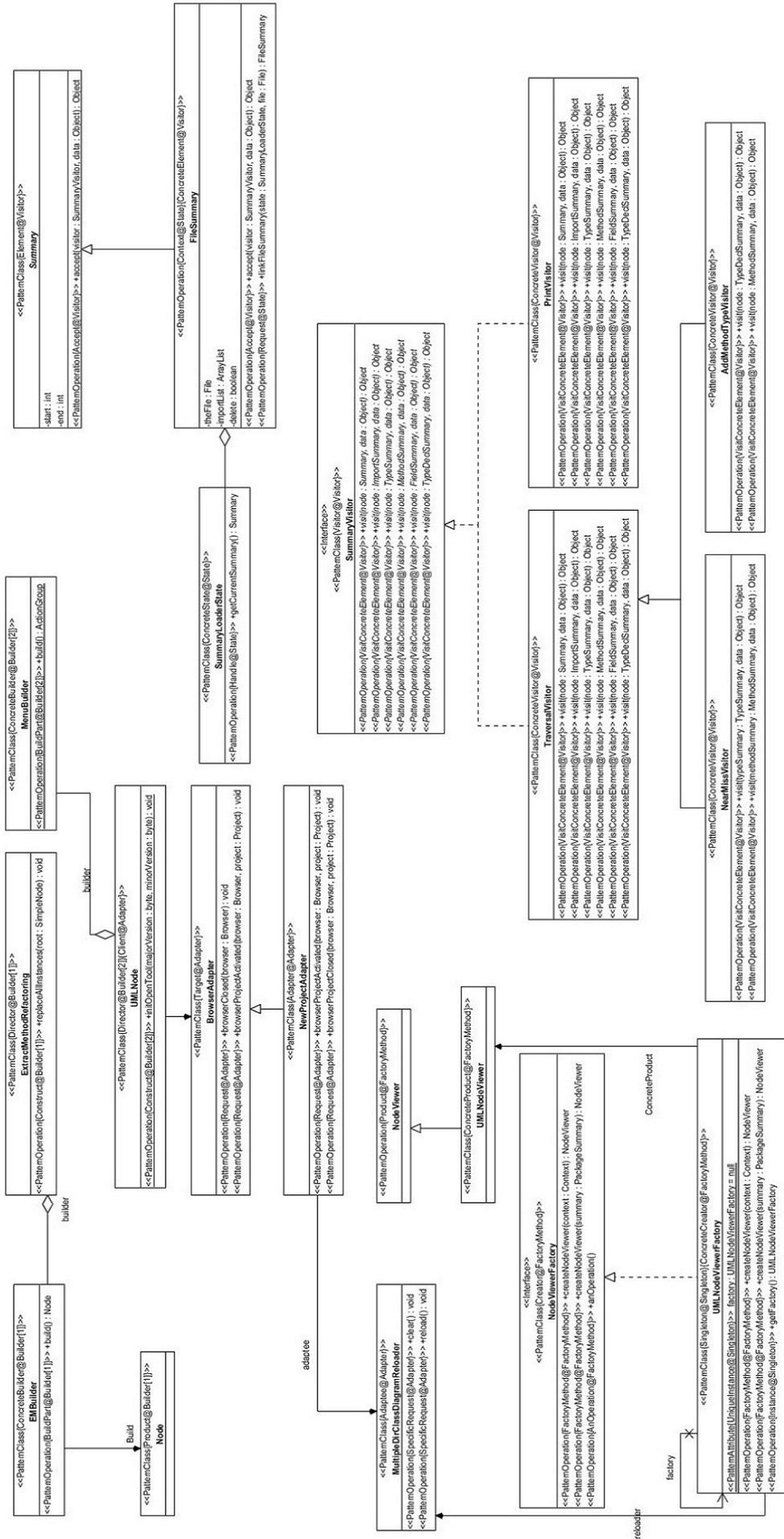


FIGURE D.4 – JRefactory / Dong

D.5 JRefactory / SPE

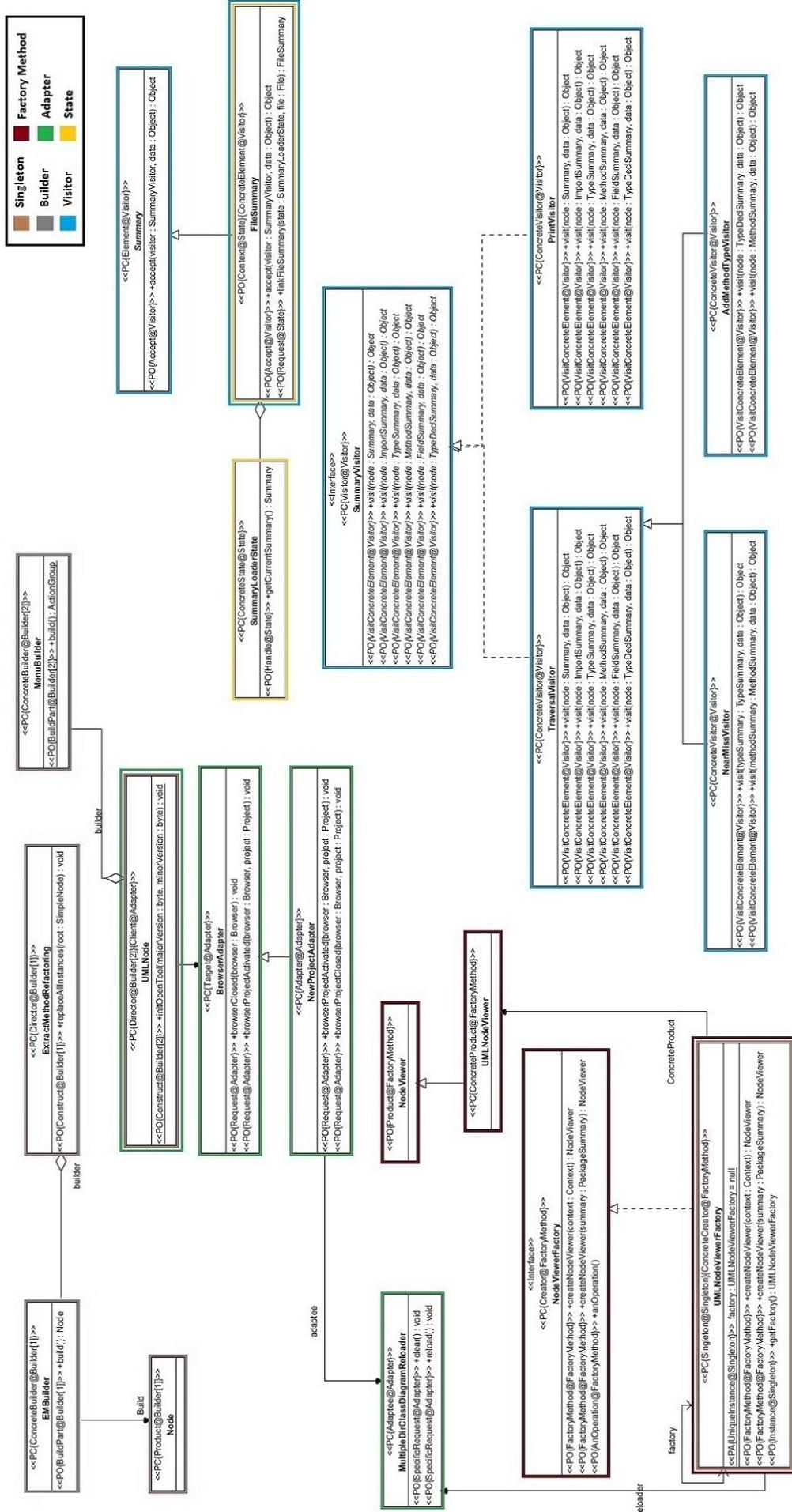


FIGURE D.5 – JRefactory / SPE

D.6 JRefactory / UMLNotation

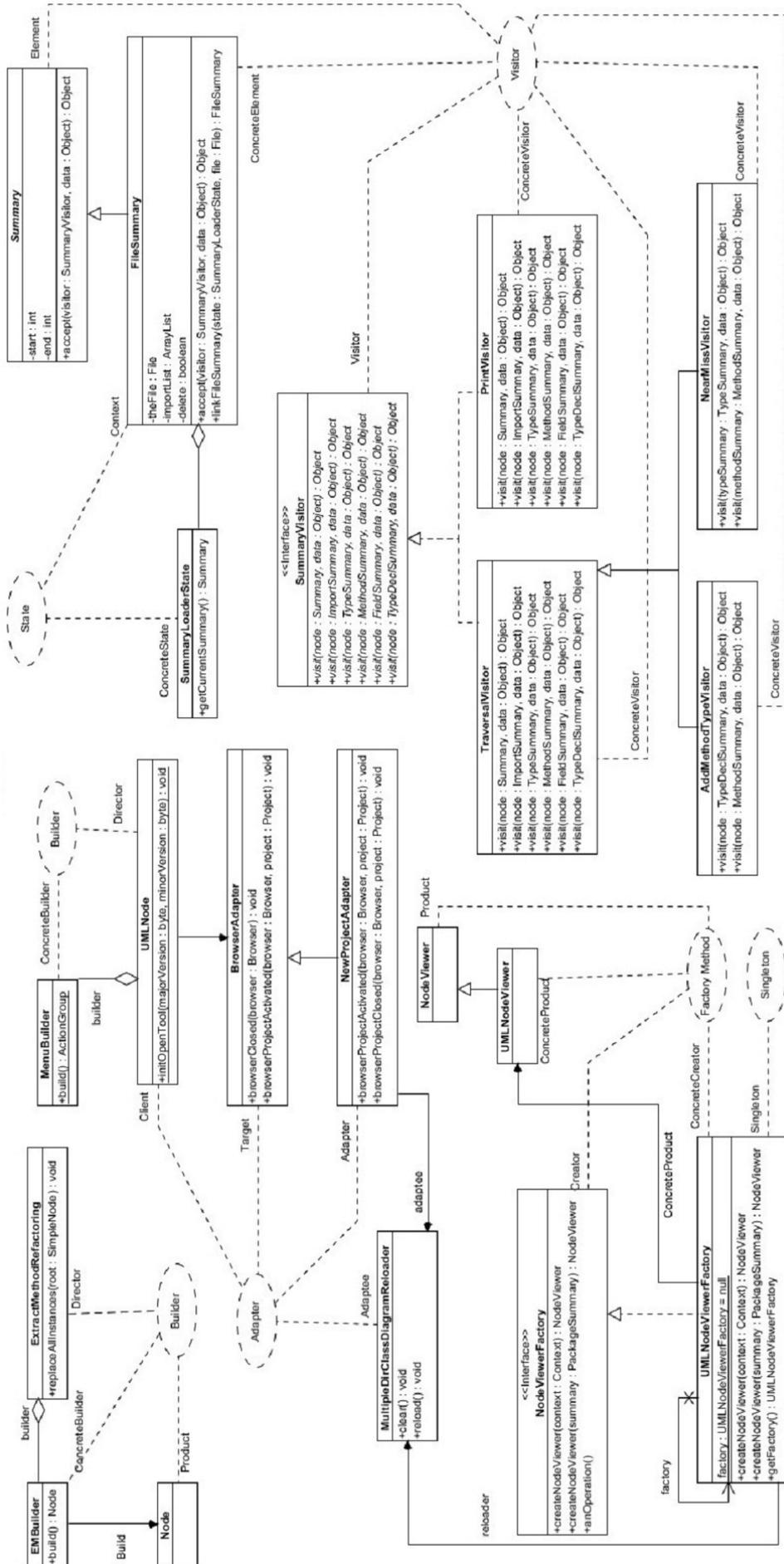


FIGURE D.6 – JRefactory / UMLNotation