

Direction des bibliothèques

AVIS

Ce document a été numérisé par la Division de la gestion des documents et des archives de l'Université de Montréal.

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

This document was digitized by the Records Management & Archives Division of Université de Montréal.

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal

Analyse, à l'aide d'oculomètres, de techniques de visualisation UML de
 patrons de conception pour la compréhension de programmes

par
Gerardo Cepeda Porras

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Aout, 2008

© Gerardo Cepeda Porras, 2008.



Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

**Analyse, à l'aide d'oculomètres, de techniques de visualisation UML de
 patrons de conception pour la compréhension de programmes**

présenté par:

Gerardo Cepeda Porras

a été évalué par un jury composé des personnes suivantes:

Pierre Poulin
président-rapporteur

Yann-Gaël Guéhéneuc
directeur de recherche

Sylvie Hamel
membre du jury

Mémoire accepté le

RÉSUMÉ

Les patrons de conception sont reconnus dans la communauté du génie logiciel comme des solutions utiles pour améliorer la qualité des programmes. Une bonne visualisation des patrons de conception est donc utile pour comprendre de façon efficace le fonctionnement d'un système. Actuellement la représentation standard pour visualiser les patrons de conception est celle de la notation de collaboration UML. Plusieurs auteurs ont remarqué des lacunes dans cette représentation et ont proposé de nouvelles représentations pour remplacer le standard. Aucun de ces auteurs n'a fait d'études empiriques pour comparer ces représentations à la représentation standard. Dans le cadre de cette maîtrise, nous avons réalisé une étude empirique pour collecter des données sur les performances des ingénieurs logiciels dans le but d'évaluer l'impact de trois représentations visuelles sur la compréhension de patrons de conception et les comparer avec la représentation standard. Nous avons utilisé des oculomètres pour mesurer la charge cognitive des sujets pendant l'exécution de l'expérience. L'analyse des données collectées montre qu'effectivement pour certaines tâches il existe une représentation qui est plus efficace que la représentation standard.

Mots clés : oculométrie, patrons de conception, visualisation, étude empirique, diagrammes de classes UML.

ABSTRACT

Design patterns are recognized in the software engineering community as useful solutions to improve the quality of programs. Thus, a good visualization of design patterns is useful to understand effectively how a system works. Currently, the standard representation to visualize design patterns is the UML Collaboration Annotation. Several authors noticed gaps in this representation and proposed new representations to replace the standard. However, none of these authors conducted any empirical study to compare these representations with the standard. In the context of this master thesis, we conducted an empirical study to collect data on the performance of software engineers to evaluate the impact of three visual representations on design pattern comprehension and to compare them with the standard. We used eye-trackers to measure the cognitive load of the subjects during the execution of the experiment. Collected data show that for certain tasks there exists a representation that is more effective than the standard representation.

Keywords: eye-tracking, design patterns, visualization, empirical study, UML class diagrams.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES TABLEAUX	vii
LISTE DES FIGURES	viii
LISTE DES SIGLES	x
DÉDICACE	xi
REMERCIEMENTS	xii
CHAPITRE 1 : INTRODUCTION	1
1.1 Contexte	1
1.2 Motivation	2
1.3 Contributions	3
1.4 Structure du mémoire	5
CHAPITRE 2 : ÉTAT DE L'ART	6
2.1 Compréhension de programme	6
2.2 Représentation visuelle de patrons pour la compréhension de programme	9
2.3 Oculométrie	18
2.4 Conclusion	20
CHAPITRE 3 : CONCEPTION DE L'EXPÉRIENCE	21
3.1 Représentations retenues pour l'expérience	22

3.2	Hypothèses	22
3.3	Sélection des objets	23
3.4	Sélection des variables indépendantes	25
3.5	Sélection des variables dépendantes	27
3.6	Sélection des sujets	28
3.7	Questions et stimulus	30
3.8	Équipement	31
3.9	Procédure	34
CHAPITRE 4 : ANALYSE ET RÉSULTATS		37
4.1	Analyse de données pour les diagrammes de 15 classes	37
4.1.1	Analyse de données pour la tâche de participation de classes : P	38
4.1.2	Analyse de données pour la tâche de composition de patrons : C	40
4.1.3	Analyse de données pour la tâche d'identification de rôles : R	41
4.1.4	Analyse de l'impact des variables d'atténuation	42
4.2	Analyse de données pour les diagrammes de 40 classes	45
4.2.1	Analyse de données pour la tâche de participation de classes : P	45
4.2.2	Analyse de données pour la tâche de composition de patrons : C	46
4.2.3	Analyse de données pour la tâche d'identification de rôles : R	48
4.2.4	Analyse de l'impact des variables d'atténuation	49
4.3	Conclusion	50
CHAPITRE 5 : MENACES À LA VALIDITÉ		54
5.1	Validité interne	54
5.2	Validité de construction	55
5.3	Validité externe	56
5.4	Validité de conclusion	57
CHAPITRE 6 : CONCLUSION		59
BIBLIOGRAPHIE		62

LISTE DES TABLEAUX

3.1	Distribution de type mixte (between-within) de sujets pour les diagrammes de 15 classes	29
3.2	Distribution de type mixte (between-within) de sujets pour les diagrammes de 40 classes	29
3.3	Distribution de sujets pour l'expérience	30
3.4	Questions	30
4.1	Données collectées	38
4.2	Effet des représentations D, G, S et U pour la tâche P, diagramme de 15 classes	39
4.3	Effet des représentations D, G, S et U pour la tâche C, diagramme de 15 classes	41
4.4	Effet des représentations D, G, S et U pour la tâche R, diagramme de 15 classes	43
4.5	Impact du niveau de connaissance de patrons de conception, diagramme de 15 classes	44
4.6	Effet des représentations D, G, S et U pour la tâche P, diagramme de 40 classes	46
4.7	Effet des représentations D, G, S et U pour la tâche C, diagramme de 40 classes	47
4.8	Effet des représentations D, G, S et U pour la tâche R, diagramme de 40 classes	49
4.9	Impact du niveau de connaissance de patrons de conception, diagramme de 40 classes	51

LISTE DES FIGURES

2.1	Représentation de patrons de conception avec des diagrammes de type Venn	10
2.2	Représentation de patrons de conception avec la notation de collaboration UML	11
2.3	Représentation de patrons de conception avec la notation de “patron :rôle”	12
2.4	Représentation de patrons de conception avec la notation de diagrammes de classe augmenté avec de patrons de conception de Schauer	13
2.5	Système Lexi [SK98] : Diagramme d’analyse de patrons ; diagramme de contexte (fenêtre en haut), diagrammes de structure (fenêtre inférieure gauche), information supplémentaire (fenêtre inférieure droite)	14
2.6	Diagramme de participation de classes de Trese et Tilley [TT07]	15
2.7	Diagramme de classes UML avec profil étendu de Dong	17
3.1	JHotDraw diagramme de 15 classes utilisé pour l’expérience	24
3.2	Distribution de questions pour l’expérience	31
3.3	Extrait de stimulus pour l’expérience - Représentation de Gamma (G)	32
3.4	Bandeau de l’oculomètre	33
3.5	Composants de l’oculomètre	33
3.6	Équipement utilisé pour faire l’expérience	35
4.1	Distribution de données pour P, diagramme de 15 classes	40
4.2	Distribution de données pour C, diagramme de 15 classes	42
4.3	Distribution de données pour R, diagramme de 15 classes	44
4.4	Impact combiné de connaissance de patrons de conception et représentations, diagramme de 15 classes	45
4.5	Distribution de données pour P, diagramme de 40 classes	47

4.6	Distribution de données pour C, diagramme de 40 classes	48
4.7	Distribution de données pour R, diagramme de 40 classes	50
4.8	Impact combiné de connaissance de patrons de conception et représentations, diagramme de 40 classes	51

LISTE DES SIGLES

D	Notation augmentée de stéréotypes de Dong et al.
DMF	Durée Moyenne de Fixations
G	Notations de type "patron :rôle" de Gamma et al.
GoF	Gang of Four, surnom donné au livre de Design Patterns écrit par Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides
PCR	Pourcentage de Réponses Correctes
RFOA	Ratio de Fixations "On_target :All_target"
RTFOA	Ratio du Temps de Fixation "On_target :All_target"
S	Diagramme de classes augmenté de patrons de conception de Schauer et al.
U	Notation de collaboration UML
ZDI	Zone d'Intérêt
ZDR	Zone de Regard

À ma famille

REMERCIEMENTS

J'aimerais tout d'abord remercier mon directeur Yann-Gaël Guéhéneuc pour son soutien, ses bons conseils et sa bonne humeur tout au long de ma recherche. Je remercie aussi Houari Sahraoui qui a toujours été disponible pour répondre à mes questions. Je voudrais remercier mes collègues et amis de l'équipe Ptidej pour leurs conseils, pour les bonnes conversations autour d'un bon café et pour leur appuie. Finalement, je remercie les membres de ma famille, José Gerardo Cepeda Torres, Maria de los Angeles Porras, Angie Cepeda, Raul Estrada, Rodolfo Beltran et Teresa Porras, ainsi que mes amis de longue date pour leurs nombreux encouragements durant mes études de maîtrise.

CHAPITRE 1

INTRODUCTION

1.1 Contexte

Les patrons de conception [GHJV98] sont des solutions à des problèmes récurrents pour la conception de logiciel dans le paradigme orienté objet. Ils sont le résultat de bonnes pratiques de conception de logiciels. La connaissance et le bon emploi de ceux-ci sont reconnus dans la communauté comme très utiles pour améliorer la qualité ainsi que la compréhension et la maintenance des programmes [GHJV98,ST02,ACC+07]. Une bonne compréhension de programme est nécessaire pour réaliser de façon efficace les tâches de développement et maintenance, car elles demandent de se faire une représentation mentale de l'architecture du programme [KKB+98,Mal08].

Les diagrammes sont des outils visuels fort utiles pour construire une telle représentation mentale en soulignant les informations les plus pertinentes sur les objets (les classes dans un contexte orienté objet) et leurs relations [CK05]. Plus particulièrement dans le domaine du génie logiciel, les diagrammes de classes UML sont des outils qui servent à diminuer l'effort cognitif requis pour aboutir à la compréhension du programme.

Actuellement la représentation standard pour visualiser les patrons de conception est celle de la notation de collaboration UML. Plusieurs auteurs ont remarqué des lacunes dans cette représentation et ont proposé de nouvelles représentations pour remplacer le standard. Aucun de ces auteurs n'a fait d'études empiriques pour comparer ces représentations au standard. La conception et la réalisation d'une étude empirique pour évaluer l'efficacité des représentations visuelles de patrons de conception sur des diagrammes de classes UML est donc une contribution concrète

dans la compréhension de programme.

1.2 Motivation

La représentation visuelle la plus utilisée pour décrire l'architecture d'un système est celle des diagrammes de classes UML [Gro01]. UML est ainsi devenu le standard pour décrire les programmes orientés objets. Plusieurs travaux ont déjà été réalisés sur des diagrammes de classes UML pour faciliter la compréhension de programme [PCA02, EvG03, SW05] et il existe aussi plusieurs outils qui génèrent ce type de diagrammes.

Plusieurs auteurs ont proposé de nouvelles représentations pour visualiser de façon efficace les patrons de conception argumentant qu'il y avait des lacunes dans la représentation standard. Ces représentations varient de fortement visuelles comme celle de Schauer et al. [SK98] à fortement textuelles comme celle de Dong et al. [DYZ07]. Nous pouvons diviser ces travaux en deux grands groupes : les approches non-UML, par exemple [EYG97, MHG02] et les approches UML [Vli98, LK98, SK98, FKGS04, TT07, DYZ07]. Nous pouvons encore diviser les approches UML en deux groupes : les représentations basées sur un seul diagramme [Vli98, SK98, DYZ07, TT07] et les approches avec plusieurs diagrammes [LK98, FKGS04]. Dans ce mémoire, nous avons choisi seulement les représentations UML basées sur un seul diagramme, car l'utilisation d'un seul diagramme est le plus courant à l'heure actuelle.

Même si plusieurs travaux ont été réalisés pour proposer de nouvelles approches pour visualiser les patrons de conception, à notre connaissance, aucun auteur n'a fait d'études empiriques pour prouver leurs améliorations.

1.3 Contributions

Dans ce projet, à partir de l'analyse de l'état de l'art, nous avons retenu 3 représentations à comparer avec celle de la notation de collaboration UML (U). Ces représentations sont : le diagramme de classes augmenté de patrons de conception (S) [SK98] (fortement visuelle), la notation augmentée de stérotypes (D) [DYZ07] (fortement textuelle) et la notation de type "patron : rôle" (G) [Vli98] (à la fois visuelle et textuelle).

Nous avons réalisé une étude empirique pour collecter des données sur les performances des ingénieurs logiciels pendant l'exécution de trois tâches de base dans la compréhension de patrons de conception : la tâche de participation de classes (c'est-à-dire, identifier toutes les classes que participent à un patron de conception), la tâche de composition de patrons de conception (qui sert à identifier tous les patrons de conception auxquels une classe participe) et la tâche d'identification des rôles joués par une classe.

Nous mesurons la performance en terme du pourcentage de réponses correctes de l'effort cognitif et du temps employé pour les sujets pour exécuter chaque tâche : le moins de charge cognitive et de temps employés et le plus de réponses correctes, plus la performance est meilleure. Afin de mesurer la charge cognitive de chaque sujet pendant l'exécution de l'expérience, nous utilisons des oculomètres. Les oculomètres sont des outils qui ont récemment commencé à être utilisés par la communauté du génie logiciel pour faire des études sur la compréhension de programme. Ces outils permettent à partir d'analyses sur les mouvements des yeux de tirer des conclusions sur les processus cognitifs sous-jacents des sujets [Ray98].

Toutes les représentations étudiées utilisent un même diagramme 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 [GHJV98]. Cependant, il y a de légères différences sur la mise en page, occasionnées principalement par des particularités d'implémentation du langage de programmation (p.e. l'utilisation d'interfaces et classes abstraites en Java) ou des particularités de mise en page (p.e. minimisation du croisement de arcs). Les patrons de conception utilisés dans les diagrammes sont : Composite, Prototype, Template Method, State et Singleton.

Nous avons choisi d'analyser séparément (car les complexités cognitives du graphe sont différentes) l'efficacité de chaque représentation sur des diagrammes de petite densité (15 classes) et sur des diagrammes de grande densité (40 classes). Nous considérons la densité de classes par rapport au nombre de classes occupant un espace de 1024×768 pixels (résolution de l'écran des expériences).

Nous avons collecté des données pour 24 sujets. À partir de l'analyse des données collectées nous rapportons, pour les diagrammes de 15 classes, que :

- les sujets ont exécuté significativement mieux la tâche de participation de classes dans les diagrammes avec U par rapport à D. Pour les autres deux représentations (G et S), nous n'avons pas trouvé de différences significatives dans l'effort cognitif et le temps, mais le pourcentage de réponses correctes a été amélioré aussi pour U par rapport à G et S.
- les sujets ont exécuté significativement mieux la tâche de composition de patrons dans les diagrammes avec D par rapport à U. Pour G, nous avons trouvé aussi que l'effort cognitif et le temps ont été significativement réduits par rapport à U. Cependant, le pourcentage de réponses correctes de G a été inférieur de 50 % à celui de U. Une étude postérieure a montré que, pour G, le niveau de connaissance des patrons de conception a une influence significative sur la performance des sujets. Pour S, nous n'avons pas trouvé des différences significatives par rapport à U.

- les sujets ont exécuté significativement mieux la tâche d'identification des rôles avec D par rapport à U. S a montré un effort cognitif inférieur à celui de U pour trouver les classes pertinentes à la tâche, mais un effort cognitif supérieur pour interpréter la représentation. Le pourcentage de réponses correctes étant de 0% pour S, peut-être causé par le manque d'information dans la représentation.

Pour les diagrammes de 40 classes, nous rapportons que les sujets ont exécuté significativement mieux la tâche de participation de classes dans U par rapport à D. Pour les deux autres représentations, nous n'avons pas trouvé de différences significatives. Le pourcentage de réponses correctes a été supérieur pour UML par rapport à G. S a eu le même pourcentage de réponses correctes que U. Pour les deux autres tâches, nous n'avons pas trouvé de différences significatives entre les représentations. Nous rapportons aussi que pour les diagrammes de 40 classes, la lisibilité est réduite par la grande densité de classes, qui peut avoir eu une influence sur les résultats obtenus.

1.4 Structure du mémoire

Le reste du mémoire est organisé comme suit. Le Chapitre 2 aborde la problématique de la compréhension de programme et présente un état de l'art des travaux de recherche portant sur la représentation visuelle des patrons de conception. Ce chapitre présente également des études empiriques importantes faites sur la compréhension de programme à l'aide d'oculomètres. Nous définissons la conception de l'expérience dans le Chapitre 3. Dans le Chapitre 4, nous présentons l'analyse et les résultats des données collectées. Nous évaluons la validité de nos résultats dans le Chapitre 5 et nous présentons nos conclusions dans le Chapitre 6.

CHAPITRE 2

ÉTAT DE L'ART

Ce travail de maîtrise se base sur plusieurs domaines de l'informatique. Dans une première partie, nous abordons la problématique de la compréhension de programmes et nous soulignons l'importance des représentations visuelles pour faciliter cette tâche. Dans une deuxième partie, nous présentons des travaux de recherche portant sur la représentation visuelle de patrons de conception dans des diagrammes. Bien qu'un bon nombre de travaux ont été réalisés, nous nous centrons seulement sur ceux qui portent sur la visualisation de patrons de conception dans des diagrammes de classes UML. Finalement dans une troisième partie, nous présentons les travaux les plus pertinents réalisés en matière d'études empiriques sur la compréhension de programmes à l'aide d'occulomètres.

2.1 Compréhension de programme

La compréhension de programme est un sujet de recherche important dans le domaine du génie logiciel, puisqu'elle est nécessaire pour la réalisation de tâches telles que le développement, la réutilisation, la maintenance et l'extension de logiciels, entre autres [Mal08]. Elle demande la capacité de se faire une représentation mentale de l'architecture d'un système [KKB⁺98]. Cette représentation mentale est construite à partir de la lecture du code et des diagrammes du système. Les diagrammes sont des outils visuels essentiels à la communication. En effet, l'utilisation de diagrammes diminue l'effort de compréhension et d'apprentissage, en omettant les détails non pertinents et, en même temps, en soulignant les informations plus pertinentes sur les objets et leurs relations. Chabris [CK05] précise que l'information est comprise plus facilement et plus vite si elle est transmise d'une manière qui corresponde le plus étroitement possible à notre propre représentation mentale et donc des diagrammes efficaces cherchent à faciliter la création d'un modèle mental.

La représentation visuelle la plus couramment utilisée par les ingénieurs logiciels pour décrire l'architecture d'un système est celle de diagrammes de classes UML. Les diagrammes de classes UML, servent à décrire la structure statique et le comportement global des systèmes logiciels [JW01]. Plusieurs travaux ont été faits au sujet de la compréhension de programme sur des diagrammes de classes UML. Purchase et al. [PCA02] ont fait des expériences sur des étudiants en génie logiciel, pour étudier l'effet de critères d'esthétique sur des diagrammes de classes UML, avec pour but d'identifier une liste ordonnée de qualités esthétiques préférées et, dans une étude postérieure, de tester si ces critères avaient une influence sur la performance des sujets lors de l'exécution de tâches de compréhension de programme. Ils ont fait plusieurs expériences sur des paires de diagrammes de classe, où un critère d'esthétique était évalué à la fois (par exemple, pour le critère esthétique d'héritage, évaluer la jonction des lignes avant d'arriver à la super classe, par rapport à l'utilisation d'arcs séparés). Des pourcentages sur les préférences des sujets sur un diagramme plutôt qu'un autre ont été collectés comme données quantitatives et des évaluations qualitatives sur chaque diagramme ont aussi été réalisées pour déterminer les raisons des choix des sujets. À partir de ces expériences, les auteurs ont identifié des critères esthétiques importants pour les diagrammes de classe UML, incluant la jonction de lignes d'héritage et les indicateurs directionnels.

Eichelberger et Gudenberg [EvG03] ont réalisé une étude sur les outils d'édition des diagrammes UML dans le cadre de la génération automatique des diagrammes de classes UML. Ils ont conclu que la communauté devrait faire des efforts pour aboutir à des compromis pour implémenter des principes d'esthétique de base et standards et ainsi faciliter la lisibilité et la compréhension des aspects statiques des systèmes logiciels. Dans une deuxième étude, Eichelberger [Eic03] a proposé un ensemble de critères d'esthétique pour les diagrammes de classes UML. Les critères utilisés dans cette approche, à contrairement aux critères étudiés par Purchase et

al., prennent en considération les sémantiques sous-jacentes aux diagrammes de classes. Il a ensuite, analysé la relation existant entre ces critères, les principes des interactions homme-machine et les principes de la conception orientée objet. À partir de cette analyse, l'auteur a implémenté un outil, SugiBib, qui intègre ces critères d'esthétique. L'auteur argumente que les critères qu'il a utilisés dans son étude pourraient augmenter la lisibilité des diagrammes de classes UML.

Sun et Wong [SW05] ont fait une sélection de critères pour la disposition efficace des classes dans les diagrammes de classes UML à partir de travaux réalisés dans le domaine, incluant les résultats des travaux cités auparavant de Purchase et Eichelberger. Ils ont ensuite classifié et sélectionné les critères les plus pertinents en accord avec les lois de la perception visuelle de la Gestalt, plus précisément la théorie de la perception organisationnelle et la théorie de la perception ségrégative [MF93]. Finalement, ils ont évalué l'efficacité des algorithmes de disposition des diagrammes de classes UML de deux outils commerciaux : Rational Rose et Borland Together. Ils ont conclu que les deux outils génèrent des diagrammes de qualité. Ils ont aussi fait des remarques sur des caractéristiques à améliorer pour les deux outils comme par exemple la paramétrisation de la disposition des classes selon les préférences de l'utilisateur et l'édition manuelle de la disposition de classes.

Les travaux présentés dans cette section 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. Pour le développement de ce projet de recherche, nous suivons donc les critères de ces travaux. Cependant, aucun de ces travaux ne s'occupent du problème de bien représenter les patrons de conception pour la compréhension de programme. La section suivante aborde les travaux les plus importants réalisés sur la visualisation de patrons de conception dans des diagrammes de classes UML.

2.2 Représentation visuelle de patrons pour la compréhension de programme

La visualisation de patrons de conception est utile pour comprendre de façon efficace le fonctionnement d'un système. Sans cette information, les développeurs ont des lacunes dans leur processus de compréhension du logiciel et ces lacunes nuisent à la maintenance et l'évolution du logiciel.

Un des premiers travaux sur la représentation visuelle de patrons de conception a été fait par Vlissides [Vli98]. Vlissides souligne l'importance d'identifier la présence et l'utilisation de patrons de conception dans la documentation de logiciels, en particulier dans des diagrammes de conception de logiciels, et fait une critique de trois représentations visuelles basées sur des diagrammes de classes UML. Pour exemplifier toutes les représentations à traiter dans cette section, nous reprenons le diagramme exemple utilisé par Vlissides. Le diagramme porte sur la conception d'un système orienté objet de gestion de fichiers hiérarchisés. Dans ce diagramme, le patron Composite est utilisé pour définir la structure du système de gestion de fichiers et le patron Proxy pour permettre des liens symboliques.

La représentation illustrée à la figure 2.1 est basée sur des diagrammes de Venn. Dans cette approche, toutes les classes participant à un patron de conception sont entourées d'une couleur particulière à chaque patron. L'auteur a remarqué que cette représentation fonctionne bien quand il y a peu de patrons dans un système. Cependant, quand plusieurs patrons sont présents, plus particulièrement dans le cas de classes participant à plusieurs patrons, les zones d'intersection sont difficiles à identifier. Finalement, il a ajouté qu'il est difficile d'identifier le rôle que joue chaque classe participant dans un patron de conception.

UML, avec ses notations de collaboration, offre une deuxième alternative. Dans cette représentation, les noms de patrons sont dans des ellipses pointillées et des

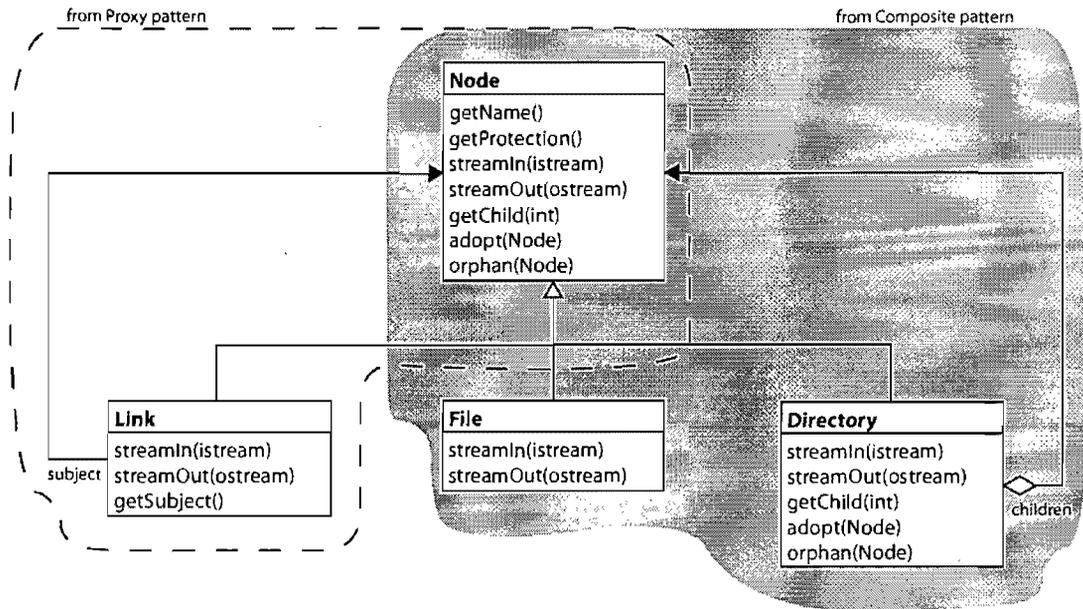


FIG. 2.1 – Représentation de patrons de conception avec des diagrammes de type Venn

lignes pointillées étiquetées sont utilisées pour indiquer le rôle joué par une classe dans un patron. À la différence de la représentation précédente, cette alternative a l'avantage de montrer de façon explicite le rôle de chaque classe (figure 2.2). 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.

Une autre approche proposée par Gamma et al. [GHJV98, Vli98] suggère une représentation avec la notation “patron :rôle”. Cette notation balise les classes avec des boîtes ombrées contenant le nom du patron et/ou le nom du rôle associé à une classe donnée (figure 2.3). Cette notation ne cause aucune interférence avec le diagramme et favorise la lisibilité car les classes donnent l'apparence d'être dans un plan différent de celui des notations. Un inconvénient de cette notation est l'augmentation de la taille du diagramme UML original. Un deuxième inconvénient est occasionné par la combinaison de la couleur grise des boîtes avec la couleur blanche

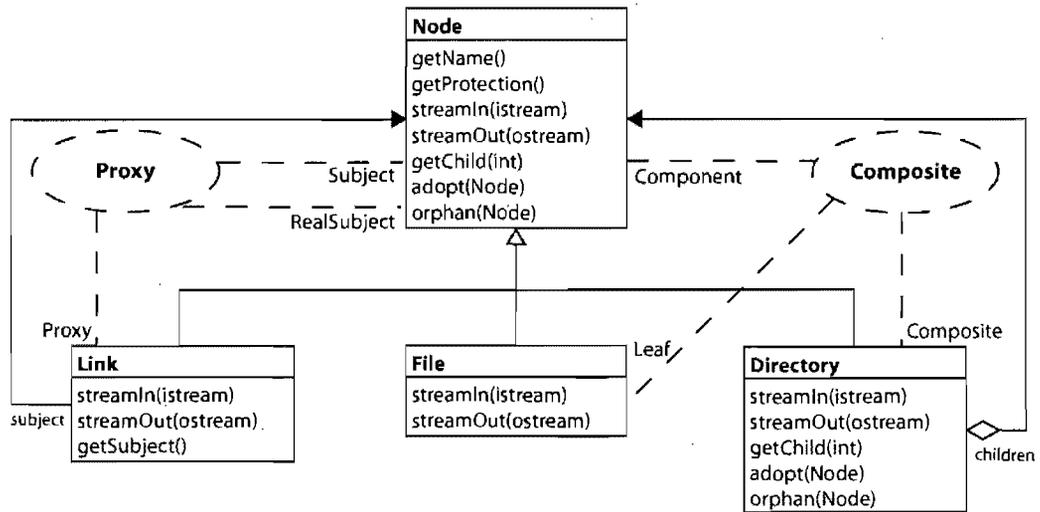


FIG. 2.2 – Représentation de patrons de conception avec la notation de collaboration UML

de la police de caractères qui peut s'avérer difficile à imprimer à basse résolution.

Schaurer et Keller [SK98] ont implémenté un prototype dans le but de faciliter la compréhension de logiciels à partir de techniques de visualisation de patrons de conception. Leur outil fait de la rétroconception de code source, a un module de dépôt de modèles de conception et de patrons de conception basé sur le GoF [GHJV98] et schématisé en UML, a un module pour la représentation visuelle de modèles et un module d'identification de patrons de conception. Pour la visualisation de patrons de conception, le prototype offre trois vues : une première montre des diagrammes de classes augmentés de patrons de conception, où des contours colorés autour de classes sont utilisés pour identifier quelles classes participent à quels patrons et la disposition de classes suit la structure typique des patrons de conception telle que montrée dans le GoF (aussi nommée "structure canonique"). La figure 2.4 montre une version du diagramme statique de cette représentation, une légende avec les noms de patrons présents dans le système a été ajoutée. Comme nous pouvons le constater sur la figure 2.4, le fait d'utiliser des contours colorés est une technique que facilite l'identification de patrons, cependant l'identification

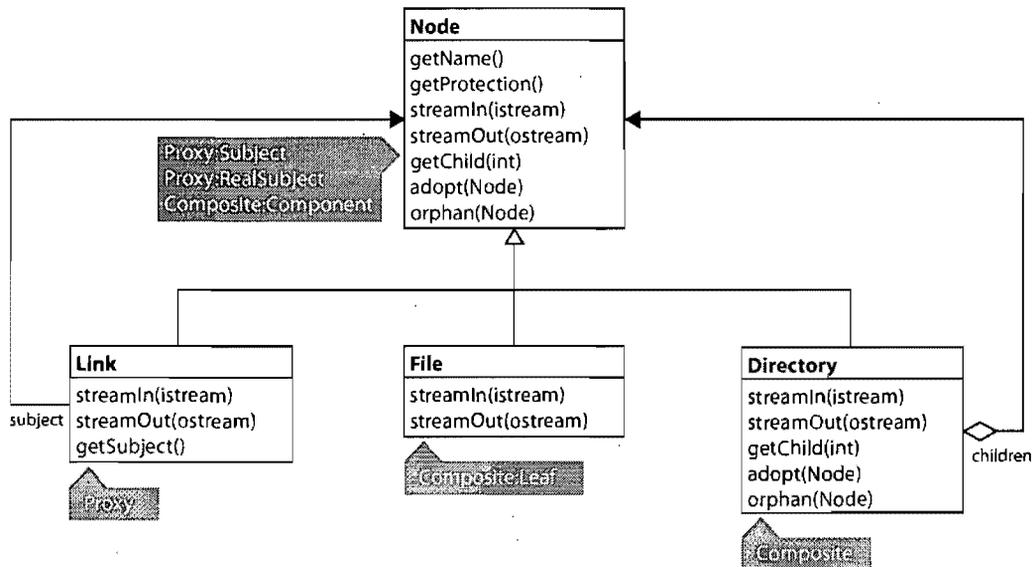


FIG. 2.3 – Représentation de patrons de conception avec la notation de “patron :rôle”

de rôles peut s’avérer difficile. Dans les prochains paragraphes, quand nous aborderons l’approche de Trese et Tilley, nous approfondirons le problème d’utiliser la représentation canonique pour visualiser les patrons de conception.

Dans une deuxième vue (ou vue d’analyse de patrons), Schauer et Keller traitent le problème d’utiliser la représentation canonique pour visualiser les patrons de conception en ajoutant des informations complémentaires. Ainsi, l’outil montre aussi le catalogue du GoF avec la structure canonique de chaque patron de conception et de l’information supplémentaire pour chaque patron comme par exemple son intention, son application, ses conséquences, etc (figure 2.5).

Dans une troisième vue, l’outil montre une représentation interactive nommée diagrammes de collaboration de patrons, laquelle est un mélange entre les diagrammes de classes et les diagrammes de paquetages UML. Dans cette vue on peut étendre ou minimiser de patrons et utiliser l’outil de zoom. L’utilisation de toutes ces vues ainsi que le catalogue du GoF dans le prototype facilitent la compréhension

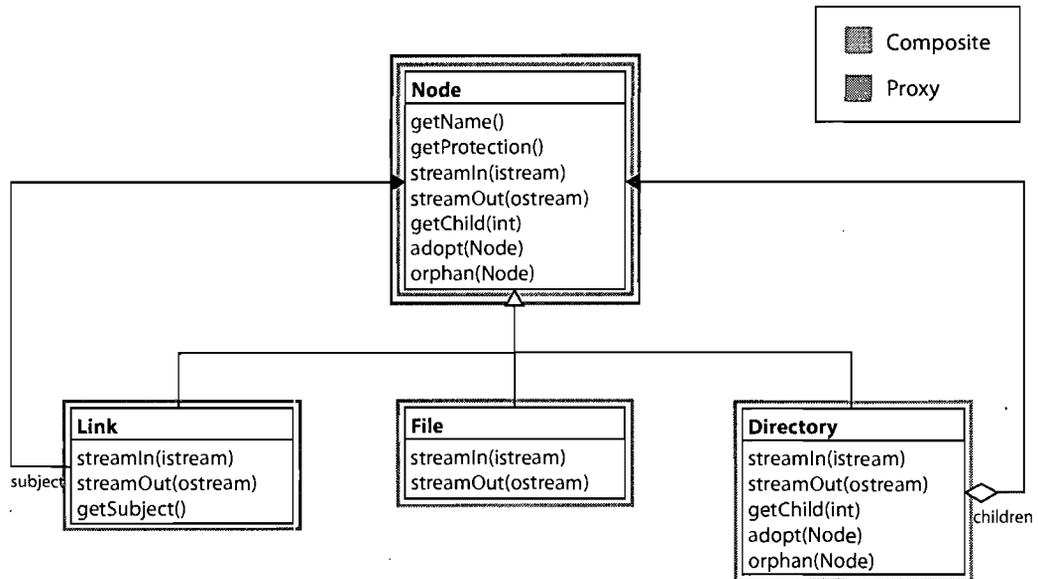


FIG. 2.4 – Représentation de patrons de conception avec la notation de diagrammes de classe augmenté avec de patrons de conception de Schauer

de patrons de conception.

Trèse et Tilley [TT07], dans le but de favoriser davantage la lisibilité et la compréhension de programme, ont proposé les diagrammes de participation de classes. Dans leur approche, un système est représenté par des ensembles de diagrammes de classes UML en groupes selon les catégories : création, comportement et structure du GoF et ensuite, sous-groupés par patron de conception dans sa représentation canonique (figure 2.6). Les auteurs soutiennent que le fait de montrer les systèmes d'une telle façon favorise la stratégie cognitive du groupage [KR98] et donc, la compréhension de l'ensemble du système. Dans leur approche, ils ont aussi proposé l'utilisation des trois niveaux de gradations de couleur grise pour produire une hiérarchie visuelle de classes (30% pour la(les) classe(s) que participe(nt) à tous les patrons de conception du diagramme, 10% pour les classes qui participent aux patrons de conception du système et 0% pour les classes auxiliaires qui ne participent à aucun patron de conception), ainsi que des métriques d'espacement de

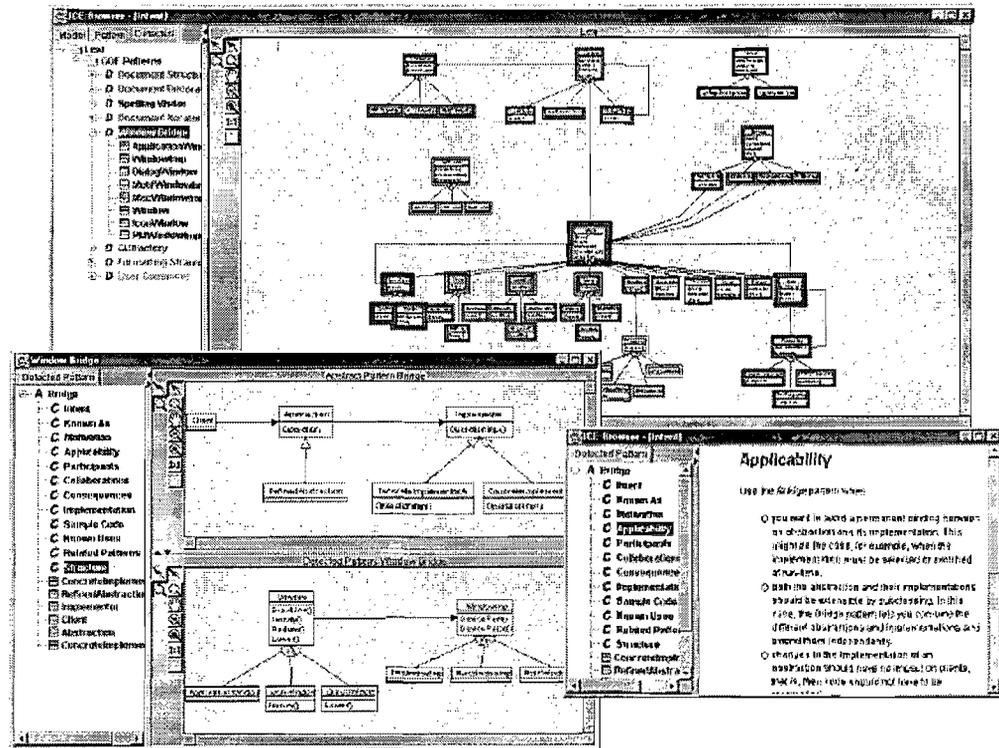


FIG. 2.5 – Système Lexi [SK98] : Diagramme d’analyse de patrons; diagramme de contexte (fenêtre en haut), diagrammes de structure (fenêtre inférieure gauche), information supplémentaire (fenêtre inférieure droite)

classes. La figure 2.6 montre un exemple de cette représentation. La principale critique à cette approche est de n’ajouter, comme informations additionnelles sur les patrons de conception, que leur représentation canonique. Il existe aussi un manque d’informations sur la représentation proposée et les diagrammes d’exemple ne sont pas assez explicites.

Nous croyons que le fait de montrer les diagrammes seulement regroupés par patron de conception, sans aucune notation additionnelle, (par exemple le nom de chaque patron de conception) peut nuire à la compréhension. Prenons l’exemple des patrons de conception État et Stratégie, ils font partie de la catégorie “comportement” du GoF, ils ont la même représentation canonique et ils utilisent normalement

Système de gestion de fichiers hiérarchisé

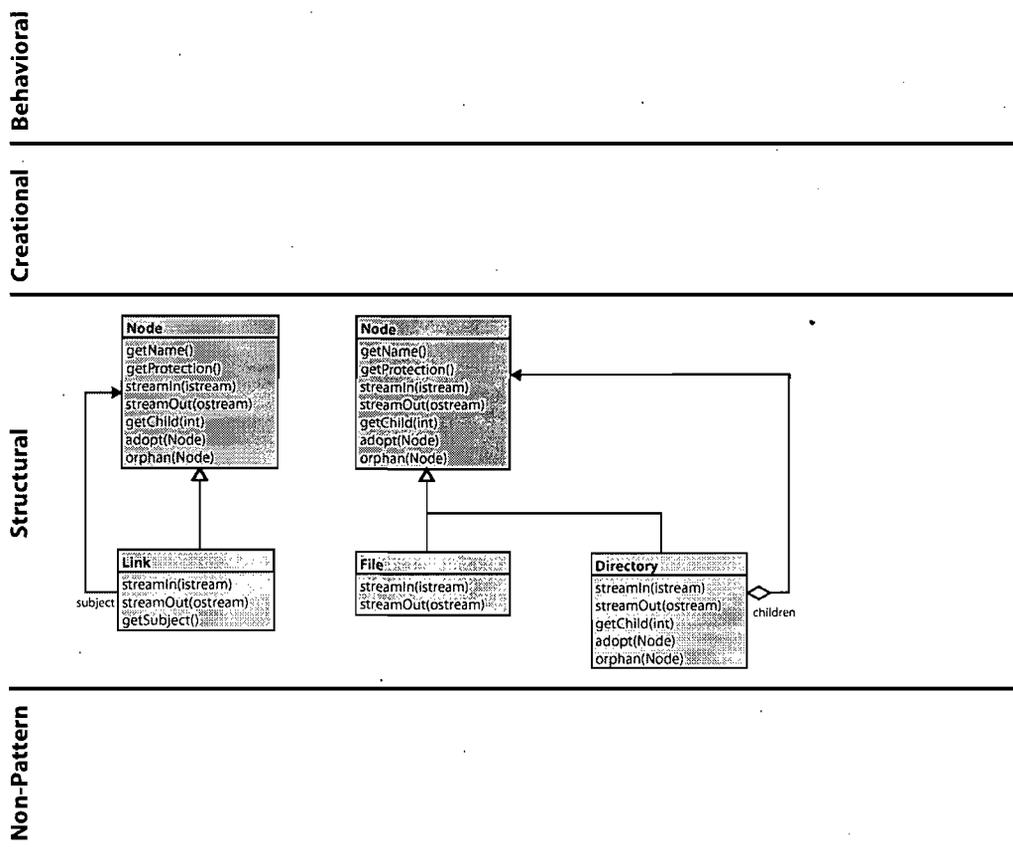


FIG. 2.6 – Diagramme de participation de classes de Trese et Tilley [TT07]

la délégation dans leurs implémentations mais leurs intentions sont différentes. L'intention du patron de conception État est de permettre à un objet de changer son comportement quand son état interne change, cela fait que l'objet agit comme s'il pouvait changer de classe. L'intention du patron de conception Stratégie est d'encapsuler des algorithmes qui vont varier indépendamment des clients qui les emploient.

Une autre critique importante à cette approche, est le tort de considérer les patrons de conception seulement comme des suggestions de mises en page. Étant donné qu'un patron est un exemple et non pas une spécification, la structure

de l'implémentation d'un patron peut varier par rapport à celle montrée dans la représentation canonique. Si nous regardons la relation Node-Link sur la figure 2.6, la tâche d'identifier de quel patron il s'agit s'avère complexe. En effet la mise en page de celui-ci a changé par rapport à la représentation canonique.

Dong et al. [DYZ07], ont proposé un profil UML avec de nouveaux stéréotypes, balises et contraintes dans le but de mieux représenter les patrons de conception dans les diagrammes UML. Dans leur approche, ils traitent de problèmes non couverts par les autres auteurs : celui de représenter explicitement les rôles que les attributs et les méthodes jouent dans un patron de conception, et celui de l'identification de multiples occurrences d'un patron, donnant une sémantique plus riche aux diagrammes de classes UML. La figure 2.7 montre un exemple concret de la notation proposée. Les stéréotypes définis sont *PatternClass*, *PatternOperation* et *PatternAttribute* pour indiquer qu'une classe, une méthode ou un attribut participent à un patron de conception (dans ce cas Proxy et Composite). Les rôles particuliers que les classes, les méthodes et les attributs jouent sont définis dans des balises de type "rôle@nom[instance]" où "nom" spécifie le nom du patron de conception, "rôle" spécifie le rôle que la classe, la méthode ou l'attribut jouent dans ce patron et "instance" spécifie à quelle instance du patron la classe, la méthode ou l'attribut appartient.

Bien que l'approche de Dong et al. s'avère intéressante, elle accroît de façon considérable la taille de chaque classe, en plus de rendre la lecture des diagrammes plus lourde en ajoutant du texte aux diagrammes. Pour résoudre ces problèmes, Dong et al. ont développé VisDP [DYZ05], un service Web pour visualiser de façon dynamique les patrons de conception sur des diagrammes de classes et de collaborations.

Les auteurs ont ensuite fait des études de complexité sur 5 représentations : UML originale sans notation de patrons, leur représentation statique avec des

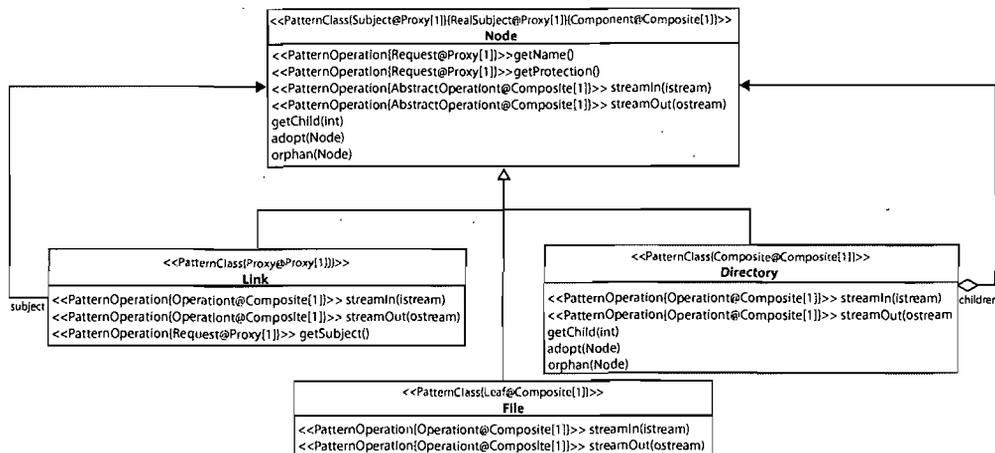


FIG. 2.7 – Diagramme de classes UML avec profil étendu de Dong

stéréotypes, leur représentation dynamique, la notation de collaborations UML et la notation de “patron :rôle” de Gamma et al. sur un système de gestion de conférences qui contient 6 patrons de conception avec 2 instances d’un patron parmi ceux-ci. Les métriques utilisées pour faire l’analyse de la complexité sont les suivantes : complexité du graphe, nombre d’arcs et de noeuds, nombre de caractères et d’unités lexicales, la métrique de complexité McCabe, la complexité de diagramme de classes et le nombre d’unités graphiques. À partir de l’analyse de ces métriques, ils ont obtenu comme résultat que leur approche statique était moins complexe que les autres approches si on ne prend pas en considération le nombre de caractères ou d’unités graphiques. Leur approche dynamique est moins ou pareillement complexe aux autres approches pour toutes les métriques. Cependant les auteurs n’ont pas fait une étude empirique sur l’efficacité et l’utilité de leur approche par rapport aux autres représentations.

Dans cette section, nous avons vu les représentations les plus importantes réalisées jusqu’à présent sur la visualisation de patrons de conception sur des diagrammes de classes UML. Les approches varient de fortement visuelles comme l’approche de Schauer et al. [SK98] à fortement textuelles comme l’approche de Dong et al. [DYZ07]. À notre connaissance aucun de ces auteurs n’a réalisé d’études

empiriques pour tester l'efficacité de leurs approches, ou pour vérifier s'il existe une différence significative entre leurs approches, et le standard des diagrammes de collaboration UML.

2.3 Oculométrie

Le but de l'oculométrie ou "eye tracking" est d'analyser le mouvement des yeux pour détecter les points d'intérêt d'une image où l'attention se porte et, à partir des données récoltées, tirer des conclusions sur les processus cognitifs sous-jacents des sujets [Ray98]. Le fondement des oculomètres est la physiologie des aptitudes visuelles humaines et des théories cognitives telles que la perception visuelle et l'attention visuelle [Duc03].

Les oculomètres ont été souvent utilisés dans le domaine de la psychologie cognitive. Cependant, de plus en plus, ce type de système est utilisé dans d'autres domaines tels que le marketing, le design industriel et l'informatique. En informatique, les études les plus souvent faites à l'aide d'oculomètres portent sur l'infographie, les interfaces homme-machine et la réalité virtuelle. Dernièrement la communauté du génie logiciel a commencé à montrer de l'intérêt pour l'utilisation de cet outil pour des études sur la compréhension de programme.

En effet, par exemple, Bednarik et Tukiainen [BT06] ont réalisé une analyse des processus cognitifs impliqués dans la compréhension de programme, le débogage et la visualisation de l'exécution de logiciels. Ils ont proposé une approche pour étudier des tendances dans des mesures répétées sur un petit ensemble de données de situations de compréhension de programme collectées par oculomètre. Les données collectées sur le mouvement des yeux ont été traitées comme des essais (trials) binomiaux. Avec cette approche, ils ont pu étudier les tendances à travers le temps, pour ensuite, caractériser le développement de stratégies et le comportement des

programmeurs pour coordonner les diverses représentations d'un logiciel pendant la compréhension de programme. Ils ont constaté que les sujets plus expérimentés analysaient en premier le code pour se former un modèle mental du logiciel, puis visualisaient l'exécution du logiciel une seule fois pour confirmer leurs hypothèses. Par contre, les sujets moins expérimentés ne commençaient pas par lire le code en premier, mais exécutaient le logiciel plusieurs fois en essayant de comprendre visuellement le fonctionnement du logiciel pour après, dans les dernières phases du processus cognitif, centrer de plus en plus leur attention sur la lecture du code.

Guéhéneuc [Gué06] a réalisé une expérience à l'aide d'oculomètres pour étudier comment les ingénieurs logiciels utilisent les diagrammes de classes pour la compréhension de programme. Dans l'expérience, les sujets ont dû répondre à deux questions de maintenance logicielle, chacune sur un système différent représenté par un diagramme de classes, pendant que les données sur le mouvement des yeux étaient enregistrées. L'auteur a défini des zones d'intérêt pour chaque classe pertinente à la réponse d'une question pour analyser quelles classes étaient l'objet de plus d'attention par les sujets pendant l'exécution de la tâche. Il a conclu sur l'importance de l'utilisation des classes et interfaces et a rapporté un apparent manque d'utilisation des relations binaires entre classes, comme l'héritage ou la composition. Cependant, il a remarqué que cet effet, a pu être causé pour la simplicité des systèmes et/ou des questions.

Yusuf et al. [YKM07] ont mené une étude similaire à celle de Guéhéneuc pour analyser l'utilisation des diagrammes de classes UML pour la compréhension de programme. Dans leur étude, ils ont analysé comment les développeurs exploiraient, parcouraient et examinaient les diagrammes de classes à partir de trois représentations différentes de diagrammes (orthogonal, three-cluster, multiple-cluster) avec des variations dans l'information sémantique, les couleurs et la disposition de classes. Dans cette expérience les sujets ont dû répondre à des questions de compréhension sur différents modules du logiciel libre d'analyse statistique Hippo-

Draw. Comme Guéhéneuc, ils ont défini des zones d'intérêt pour chaque diagramme et ont mesuré l'efficacité des représentations basées sur la moyenne du nombre total de fixations par tâche accomplie. Ils ont conclu sur le fait que la représentation avec la disposition en multiple-cluster était la plus efficace des trois. Ils ont aussi remarqué que l'utilisation d'information additionnelle telle que les stéréotypes et les couleurs rendent les diagrammes plus lisibles et que le fait d'adopter des notations visuelles moins semblables (par exemple : agrégation et généralisation) peut réduire l'effort de compréhension de programme.

2.4 Conclusion

Dans ce chapitre, nous avons fait une revue des principaux travaux reliés à l'étude empirique présentée dans ce mémoire. Nous avons d'abord présenté les principes de la compréhension de programme et les travaux plus importants réalisés jusqu'à présent pour faciliter la compréhension des diagrammes de classes UML. Ensuite, nous avons présenté les représentations plus importantes pour visualiser des patrons de conception sur des diagrammes de classes UML. Finalement, nous avons présenté les oculomètres, un outil qui prend de plus en plus d'importance dans la communauté du génie logiciel pour analyser la compréhension de programme et qu'on utilise dans notre recherche. Nous avons aussi présenté les plus importantes études empiriques réalisés avec cet outil dans le domaine de compréhension de programme.

CHAPITRE 3

CONCEPTION DE L'EXPÉRIENCE

Comme nous l'avons souligné dans le chapitre précédent, plusieurs auteurs ont suggéré de nouvelles représentations pour visualiser de façon efficace les informations relatives aux patrons de conception et ainsi faciliter la compréhension des programmes. Tous ces auteurs ont comparé leur représentation à celle de la notation de collaboration UML et ont signalé que leurs approches présentent des améliorations par rapport à ces diagrammes. Cependant, aucun auteur, à notre connaissance, n'a fait d'études empiriques pour prouver ces améliorations. Le but de notre expérience est donc d'inférer comment les ingénieurs logiciels utilisent les différentes notations ajoutées sur des diagrammes de classes UML pour repérer des informations relatives aux patrons de conception et vérifier s'il existe des différences entre ceux-ci et les diagrammes de collaboration UML.

Précisément, nous voulons tester si les hypothèses faites par Dong et al. [DYZ07], Gamma et al. [Vli98] et Schauer et al. [SK98] sont vraies, c'est-à-dire, si l'utilisation de leurs représentations apporte des améliorations à la compréhension de patrons de conception par rapport à la représentation de la notation de collaboration UML. Pour cela, nous avons demandé à des sujets d'effectuer des tâches spécifiques sur des diagrammes ayant une petite densité de classes ainsi que sur des diagrammes ayant une plus grande densité de classes (15 et 40 classes respectivement) et, avec l'aide d'un oculomètre, nous avons collecté des données pertinentes pour analyser l'effort cognitif des sujets (défini plus tard dans ce chapitre). Le reste de ce chapitre décrit en détail la conception de notre expérience.

3.1 Représentations retenues pour l'expérience

Après avoir analysé les représentations présentées dans la section antérieure, nous avons retenu 3 représentations pour les comparer avec celle de la notation de collaboration UML (U). La représentation de notations de type "patron :role" proposé par Gamma et al. (G) [Vli98] est intéressante pour notre étude, car même si elle accroît de façon significative la taille du diagramme, le fait de montrer l'information des patrons dans un autre plan peut alléger la charge cognitive de la lecture du diagramme et faciliter davantage le repérage des informations relatives aux patrons de conception. La représentation par diagrammes de classes augmentés de patrons de conception proposée par Schauer et al. (S) [SK98] a été retenue pour sa simplicité, la facilité d'identification visuelle des patrons et aussi pour vérifier l'efficacité de l'utilisation de la représentation canonique du GoF pour l'identification des informations clés sur les patrons de conception (idée aussi utilisée dans l'approche de Trese et Tilley [TT07]). Finalement, la représentation avec stéréotypes proposée par Dong et al. (D) [DYZ07] a été retenue pour sa richesse sémantique et aussi parce qu'elle est fortement textuelle.

3.2 Hypothèses

Ainsi, nous voulons évaluer les hypothèses nulles suivantes :

- H_{0_1} : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre U et S.
- H_{0_2} : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre U et G.
- H_{0_3} : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre U et D.

Si les hypothèses nulles sont rejetées, nous pourrions supposer (dans la limite des menaces pour la validité présentées dans le chapitre 5) que les hypothèses al-

ternatives suivantes sont vérifiées :

- $H_{\alpha_{1.1}}$: l'effort cognitif moyen des sujets face à S est supérieur à celui obtenu face à U.
- $H_{\alpha_{1.2}}$: l'effort cognitif moyen des sujets face à S est inférieur à celui obtenu face à U.

- $H_{\alpha_{2.1}}$: l'effort cognitif moyen des sujets face à G est supérieur à celui obtenu face à U.
- $H_{\alpha_{2.2}}$: l'effort cognitif moyen des sujets face à G est inférieur à celui obtenu face à U.

- $H_{\alpha_{3.1}}$: l'effort cognitif moyen des sujets face à D est supérieur à celui obtenu face à U.
- $H_{\alpha_{3.2}}$: l'effort cognitif moyen des sujets face à D est inférieur à celui obtenu face à U.

3.3 Sélection des objets

Nous avons choisi le logiciel avec code de source libre JHOTDRAW comme objet de notre expérience. JHOTDRAW est un cadriciel utilisé pour implanter des dessins techniques. Sa conception a été faite par Gamma et Eggenschwiler pour illustrer le bon emploi de patrons de conception.

Nous avons eu recours au module d'ingénierie inverse de Netbeans pour obtenir un diagramme de classes UML du cadriciel, car nous n'avions pas accès à sa documentation complète. Pour cette expérience, nous avons décidé d'utiliser des diagrammes avec des classes participant dans les patrons de conception suivants : Composite, Prototype, Template Method, State et Singleton, tels qu'appliqués dans

JHotDraw (leur structure change un peu par rapport à celle montrée dans leurs représentations canoniques).

Tant pour les diagrammes de 15 classes (figure 3.1) que pour les diagrammes de 40 classes (disponible sur le Web <http://www-etud.iro.umontreal.ca/~cepedapg/memoire>), les classes participant aux différents patrons de conception sont les mêmes. Il y a seulement un léger changement dans la disposition des classes pour les diagrammes de 40 classes en raison des relations entre les nouvelles classes ajoutées au diagramme. Toutes les notations à évaluer pour l'expérience, sont basées sur le même diagramme de classes UML de base. Le site cité précédemment contient toutes les représentations utilisées pour l'expérience.

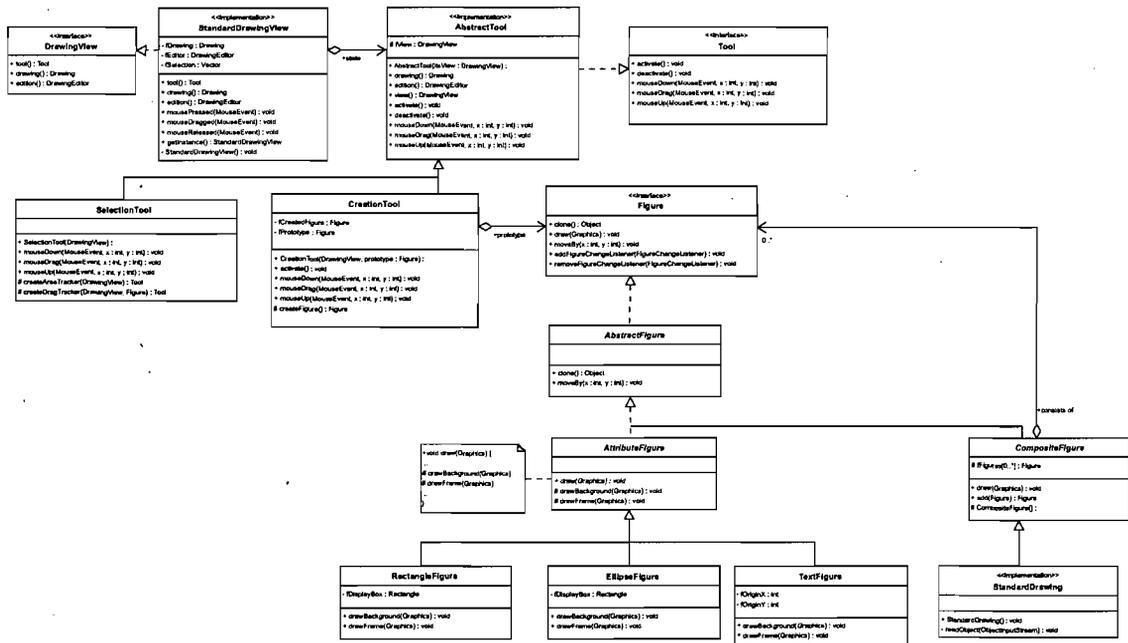


FIG. 3.1 – JHotDraw diagramme de 15 classes utilisé pour l'expérience

3.4 Sélection des variables indépendantes

À partir des hypothèses mentionnées dans la section précédente, nous avons identifié les variables indépendantes suivantes :

- **représentations** : U, S, G, D sont les valeurs possibles pour cette variable en rapport avec les quatre représentations évaluées dans notre expérience. La première représentation U est la représentation de la notation de collaboration UML. Nous utilisons G pour la représentation de notations de type “patron :rôle” de Gamma et al., S pour la représentation des diagrammes de classes augmentée de patrons de conception proposée par Schauer et al. et D pour celle de stéréotypes proposée par Dong et al. Pour différencier les diagrammes de classes de petite densité de ceux de grande densité, nous utilisons les indices 15 et 40 respectivement. Nous n’analyserons pas les diagrammes de 15 classes par rapport aux diagrammes de 40 classes, car la charge cognitive est différente ;
- **tâches** : dans l’approche cognitive de systèmes [Hut95], le plus important est de trouver un moyen pour que l’information puisse être acquise le plus vite et le plus facilement possible [War05]. Le but principal des tâches choisies est de mettre les sujets dans des situations à partir desquelles nous pourrions analyser comment les sujets utilisent les différentes notations contenues dans chaque représentation pour repérer les informations relatives aux patrons de conception. Ainsi nous pourrions comparer la charge cognitive liée à l’accomplissement de chaque tâche sur chaque représentation. Aussi, les tâches choisies pour cette expérience portent sur le repérage des informations relatives aux patrons de conception. Nous avons identifié les tâches suivantes :
 - la **participation des classes**, c’est-à-dire, identifier toutes les classes que participent à un patron de conception ;
 - la **composition de patrons**, c’est-à-dire, identifier tous les patrons de conception auxquels une classe donnée participe ;

- les **rôles joués**, c'est-à-dire, identifier les rôles que chaque classe, méthode et attribut jouent dans un patron de conception ;
- la **différentiation d'instances**, c'est-à-dire, identifier toutes les instances d'un même patron de conception.

Cependant, la seule représentation que permet toutes ces tâches est celle des stéréotypes de Dong et al. et inclure toutes ces tâches risque donc de défavoriser les autres représentations. Aussi, nous avons exclu les tâches de différenciation d'instances et l'identification des rôles de méthodes et d'attributs dans notre étude. Finalement nous retenons 3 tâches avec pour valeurs :

- P pour la participation de classes ;
- C pour la composition de patrons ;
- R pour les rôles joués par chaque classe.

Pour chaque tâche, nous testerons les hypothèses mentionnées précédemment. Dans la suite de ce mémoire, les expériences accomplies par les sujets seront nommées :

<TÂCHE>_<REPRÉSENTATION><ALTERNATIVE_NUM_CLASSES>.

Nous retenons deux variables d'atténuation pour mettre en perspective et aboutir à une meilleure compréhension des résultats obtenus :

- **connaissance de JHotDraw** : la familiarité d'un sujet avec le système JHotDraw. Le niveau est établi à partir d'un questionnaire. Les valeurs possibles sont [0,1,2] où 0 veut dire que le niveau de connaissance de JHotDraw du sujet est basique, 1 si le sujet connaît moyennement JHotDraw et 2 si le sujet a une très bonne connaissance de JHotDraw.
- **connaissance des patrons de conception** : le niveau de connaissance de patrons de conception. La valeur de cette variable est aussi établie à l'aide d'un questionnaire, comme nous l'avons fait pour la variable précédente.

3.5 Sélection des variables dépendantes

Les variables dépendantes pour cette expérience ont été choisies à partir des capacités de l'oculomètre, de nos hypothèses et de nos variables indépendantes. Nous avons classifié les variables dépendantes en deux types : mesures de performance et mesures de l'effort cognitif. Notre mesure de performance est le pourcentage de réponses correctes (PCR) pour chaque représentation pour chaque tâche.

Pour chaque diagramme de chaque représentation (S_{15}/S_{40} , D_{15}/D_{40} , G_{15}/G_{40} et U_{15}/U_{40}), nous avons défini un ensemble de zones de regard (ZDR) et de celui-ci nous avons défini des sous-ensembles de zones d'intérêt (ZDI). Une zone de regard est un élément de notation faisant partie du diagramme. Une zone d'intérêt est un élément de notation pertinent dans notre diagramme pour accomplir une tâche, c'est-à-dire, un élément du diagramme qui doit faire partie du centre d'attention du sujet. Les zones d'intérêt varient selon la tâche à accomplir. À partir des fixations enregistrées (voir section 3.8) sur les zones d'intérêt nous utiliserons les mesures suivantes :

- le ratio de fixations "on_target :all_target" RFOA [GK99] : ce ratio mesure l'efficacité de la recherche en fonction du contenu, un ratio plus petit indique une efficacité plus faible et donc un effort plus grand du sujet pour trouver les éléments pertinents pour accomplir une certaine tâche. La formule pour obtenir cette mesure est la suivante :

$$RFOA = \frac{\sum_{i=1}^{\text{TOTAL DE FIXATIONS DANS ZDI}} F_i}{\sum_{j=1}^{\text{TOTAL DE FIXATIONS DANS ZDR}} F_j}$$

où F_k représente la fixation k .

- le ratio du temps de fixation "on_target :all_target" RTFOA [GK99] : le ratio entre le temps passé dans les ZDI et le temps passé dans les ZDR.

Un ratio plus petit indique une efficacité plus faible. La formule pour obtenir cette mesure est la suivante :

$$RTFOA = \frac{\sum_{i=1}^{\text{TOTAL DE FIXATIONS DANS ZDI}} (TF(F_i) - TI(F_i))}{\sum_{j=1}^{\text{TOTAL DE FIXATIONS DANS ZDR}} (TF(F_j) - TI(F_j))}$$

où $TF(F_k)$ et $TI(F_k)$ représentent le temps final et le temps initial pour la fixation F_k .

- **la durée moyenne de fixations DMF** : cette mesure est corrélée avec les fonctions cognitives [GK99]. Des fixations plus longues indiquent un accroissement des fonctions cognitives, c'est-à-dire que le sujet est en train de faire des fixations plus longues pour l'interprétation ou l'assemblage des éléments d'une représentation [GK99, Duc03]. Les représentations qui requièrent des fixations plus longues sont donc moins efficaces que celles qui requièrent des fixations moins longues. La durée moyenne des fixations est calculée par la somme du temps de la durée de toutes les fixations dans la tâche effectuée divisée par le nombre de fixations :

$$DMF = \frac{\sum_{i=1}^{\text{NOMBRE DE FIXATIONS}} (TF(F_i) - TI(F_i))}{\#\{Fixations\}}$$

où $TF(F_i)$ et $TI(F_i)$ représentent le temps final et le temps initial pour la fixation F_i .

3.6 Sélection des sujets

L'expérience a été faite avec 24 sujets bénévoles faisant des études de deuxième ou troisième cycle au département d'informatique et de recherche opérationnelle

de l'Université de Montréal. Tous les étudiants ont déjà fait des projets durant leurs études où ils devaient modéliser des systèmes logiciels et ils ont une bonne expérience de l'utilisation des diagrammes de classes UML. Tous les étudiants avaient aussi des connaissances sur les patrons de conception. Les sujets ont été placés dans des groupes balancés. Des groupes balancés permettent de simplifier et de donner plus de force à notre analyse statistique pour les données collectées [WRH⁺00]. La table 3.1 montre la distribution des sujets pour les diagrammes de 15 classes et la table 3.2 de 40 classes où **G** et **G'** sont des groupes formés à partir du même ensemble de 24 sujets.

	<i>C</i>	<i>P</i>	<i>R</i>
D	G_1	G_1	G_1
G	G_2	G_2	G_2
S	G_3	G_3	G_3
U	G_4	G_4	G_4

TAB. 3.1 – Distribution de type mixte (between-within) de sujets pour les diagrammes de 15 classes

	<i>C</i>	<i>P</i>	<i>R</i>
D	G'_1	G'_1	G'_1
G	G'_2	G'_2	G'_2
S	G'_3	G'_3	G'_3
U	G'_4	G'_4	G'_4

TAB. 3.2 – Distribution de type mixte (between-within) de sujets pour les diagrammes de 40 classes

Chaque sujet a fait l'expérience pour 2 représentations différentes avec des densités de classes différentes. Le but de la création de blocs d'individus par groupe de représentations ainsi que la randomisation des tâches (voir la figure 3.3) est de minimiser les effets d'apprentissage [WRH⁺00, Duc03]. La table 3.3 montre la

distribution finale des sujets pour l'exécution de l'expérience.

	15	40
D	$S_9, S_{11}, S_{12}, S_{21}, S_{23}, S_{24}$	$S_3, S_5, S_6, S_{13}, S_{17}, S_{18}$
G	$S_6, S_8, S_{10}, S_{18}, S_{20}, S_{22}$	$S_2, S_4, S_{12}, S_{14}, S_{16}, S_{24}$
S	$S_4, S_5, S_7, S_{16}, S_{17}, S_{19}$	$S_1, S_{10}, S_{11}, S_{13}, S_{22}, S_{23}$
U	$S_1, S_2, S_3, S_{13}, S_{14}, S_{15}$	$S_7, S_8, S_9, S_{19}, S_{20}, S_{21}$

TAB. 3.3 – Distribution de sujets pour l'expérience

3.7 Questions et stimulus

Les questions choisies pour les tâches définies dans la section 3.4 sont montrées dans la table 3.4. Nous avons défini 2 questions pour chaque type de tâche. Les questions ont été choisies pour que l'information montrée dans les diagrammes soit suffisante pour trouver la réponse en 1-2 minutes. La figure 3.2 montre la distribution des questions pour l'expérience.

Tâche de composition de patrons : C
Q1. Nommer tous les patrons de conception dans lesquels participe la classe CreationTool.
Q2. Nommer tous les patrons de conception dans lesquels participe la classe AttributeFigure.
Tâche de participation de classes : P
Q5. Nommer toutes les classes faisant partie du patron Composite.
Q6. Nommer toutes les classes faisant partie du patron Stat.
Tâche d'identification de rôles joués par une classe : R
Q3. Nommer tous les rôles joués par la classe AbstractFigure.
Q4. Nommer tous les rôles joués par la classe StandardDrawingView.

TAB. 3.4 – Questions

S1	D15,UML	Q1	Q4	Q5
	D40,Schauer	Q3	Q2	Q5
S2	D15,UML	Q3	Q2	Q5
	D40,Gamma	Q1	Q4	Q5
S3	D15,UML	Q4	Q2	Q5
	D40,Dong	Q5	Q3	Q1
S4	D15,Schauer	Q5	Q1	Q4
	D40,Gamma	Q2	Q5	Q3
S5	D15,Schauer	Q5	Q3	Q2
	D40,Dong	Q4	Q1	Q6
S6	D15,Gamma	Q5	Q1	Q3
	D40,Dong	Q2	Q4	Q7
S13	D15,UML	Q5	Q2	Q4
	D40,Schauer	Q3	Q5	Q1
S14	D15,UML	Q5	Q4	Q2
	D40,Gamma	Q1	Q3	Q6
S15	D15,UML	Q4	Q2	Q5
	D40,Dong	Q3	Q5	Q1
S16	D15,Schauer	Q5	Q1	Q4
	D40,Gamma	Q2	Q5	Q3
S17	D15,Schauer	Q2	Q4	Q5
	D40,Dong	Q5	Q1	Q3
S18	D15,Gamma	Q3	Q5	Q1
	D40,Dong	Q4	Q2	Q5
S7	D40,UML	Q1	Q4	Q6
	D15,Schauer	Q3	Q2	Q5
S8	D40,UML	Q5	Q1	Q3
	D15,Gamma	Q2	Q4	Q5
S9	D40,UML	Q5	Q3	Q2
	D15,Dong	Q4	Q1	Q5
S10	D40,Schauer	Q3	Q2	Q5
	D15,Gamma	Q1	Q4	Q5
S11	D40,Schauer	Q4	Q2	Q5
	D15,Dong	Q5	Q3	Q1
S12	D40,Gamma	Q5	Q3	Q2
	D15,Dong	Q4	Q1	Q5
S19	D40,UML	Q4	Q1	Q5
	D15,Schauer	Q5	Q3	Q2
S20	D40,UML	Q1	Q4	Q5
	D15,Gamma	Q3	Q2	Q5
S21	D40,UML	Q3	Q2	Q5
	D15,Dong	Q5	Q4	Q1
S22	D40,Schauer	Q5	Q1	Q3
	D15,Gamma	Q4	Q5	Q2
S23	D40,Schauer	Q1	Q5	Q3
	D15,Dong	Q4	Q2	Q5
S24	D40,Gamma	Q4	Q1	Q5
	D15,Dong	Q2	Q3	Q5

FIG. 3.2 – Distribution de questions pour l'expérience

Dans le domaine de l'oculométrie, l'objet regardé par un sujet est nommé "stimulus". Nous avons combiné une question et le diagramme correspondant dans un même stimulus. Plusieurs recherches ont montré la tendance des sujets à regarder le coin supérieur-gauche d'un stimulus [Boj05, GSL⁺02]. Pour minimiser l'impact de cette tendance, nous plaçons la question dans la partie supérieure de l'écran et le diagramme occupe la partie restante. La figure 3.3 montre un exemple de stimulus.

3.8 Équipement

Pour cette expérience, nous utilisons l'oculomètre EyeLink II de SR Research. Ce système est le plus rapide et a la plus haute définition de tous les oculomètres de type "monté sur la tête" avec un débit de 500 échantillons par second et une résolution de bruit inférieure à 0.01°. Il a aussi une très bonne précision avec un taux moyen d'erreur du regard fixe inférieur à 0.5° [Eye08].

L'oculomètre est composé de deux ordinateurs, l'un utilisé pour l'exécution des expériences et l'autre utilisé par l'expérimentateur pour configurer le système et faire le traitement des données. Les deux ordinateurs communiquent entre eux au moyen d'une connexion Ethernet. Le système comporte aussi un bandeau avec

Nommer tous les patrons de conception dans lesquels participe la classe CreationTool

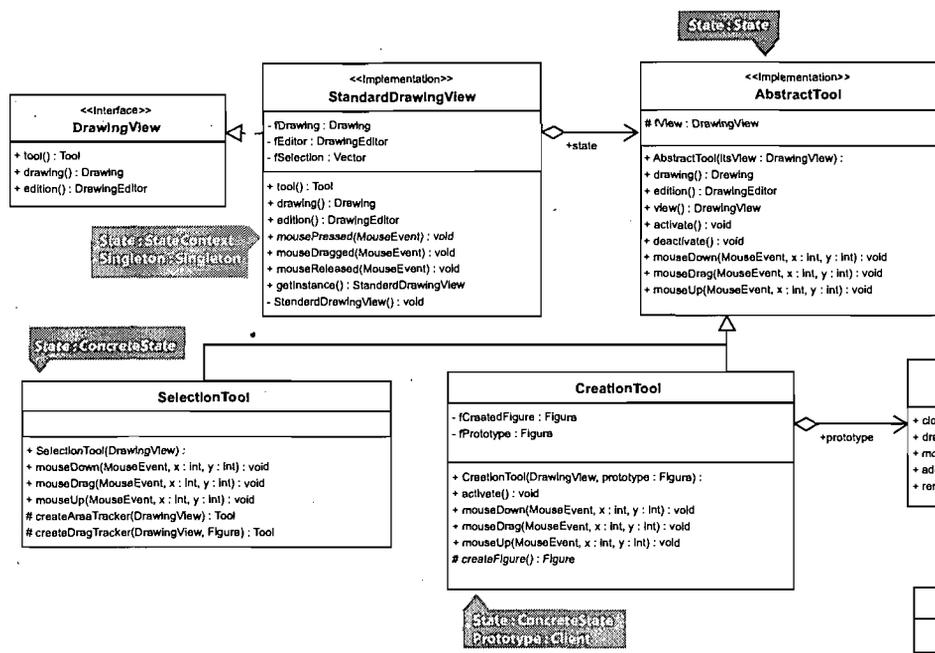


FIG. 3.3 – Extrait de stimulus pour l'expérience - Représentation de Gamma (G)

deux caméras et des émetteurs de rayons infra-rouge (figure 3.4). Les rayons infra-rouge sont utilisés par les caméras après avoir été réfléchés sur la cornée des sujets pour détecter les mouvements des yeux. Sur l'ordinateur des expériences, quatre capteurs détectent la position du bandeau par rapport à l'écran à partir des signaux envoyés par l'émetteur de rayon infra-rouge central situé sur le bandeau. Les caméras, en combinaison avec les capteurs, permettent à l'oculomètre d'enregistrer avec précision la position du regard des sujets sur l'écran.

La communication entre les deux ordinateurs est basée sur le principe de "Action / Événement" → "Réaction". Quand l'ordinateur des expériences émet un événement, l'ordinateur de l'expérimentateur réagit et reprend le contrôle. Essentiellement, l'ordinateur des expériences envoie la position du regard du sujet à l'ordinateur de l'expérimentateur pour que celui-ci la sauvegarde sur son disque dur en

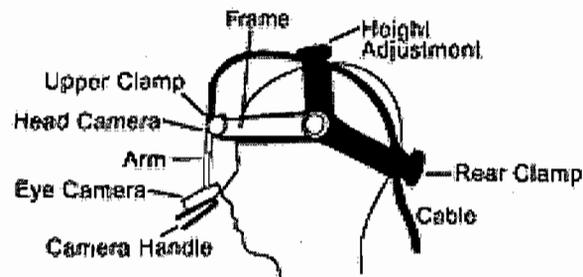


FIG. 3.4 – Bandeau de l'oculomètre

temps réel. Une fois l'expérience finie, l'ordinateur de l'expérimentateur renvoie le fichier des données complètes à l'ordinateur des expériences pour qu'il puisse être analysé plus tard. La figure 3.5 montre tous les composants de l'oculomètre.

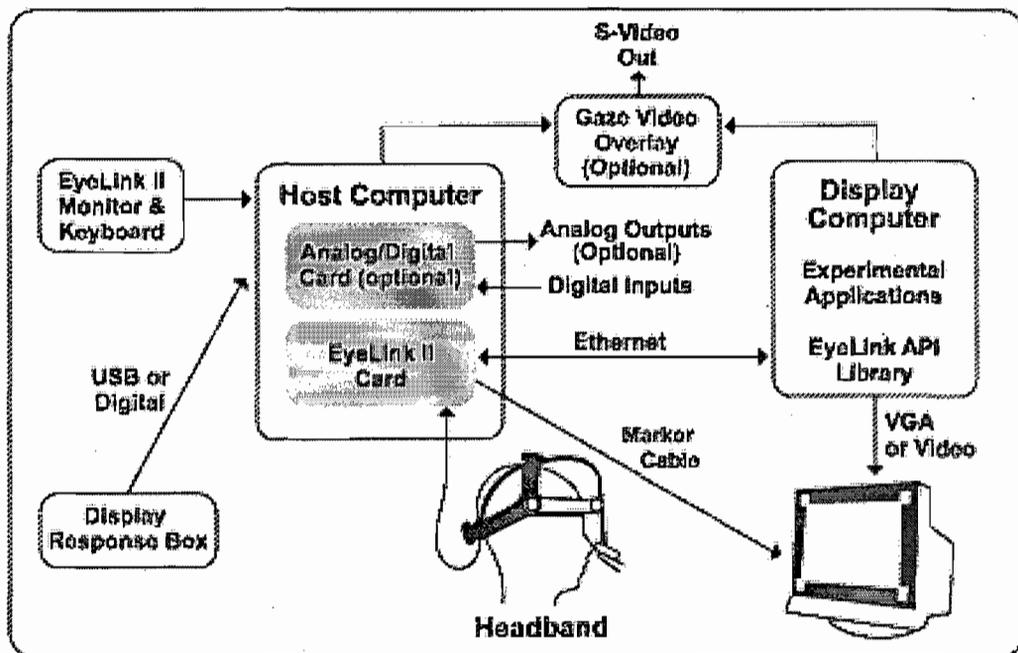


FIG. 3.5 – Composants de l'oculomètre

Le système peut collecter des données de deux types : positions brutes et positions traitées (analysées). Les positions traitées sont données en termes de fixations et de saccades basées sur des seuils physiologiques. Une fixation est la stabilisation

de l'oeil sur une zone d'intérêt pour une certaine période de temps tandis que les saccades sont des mouvements rapides de l'oeil pour focaliser sur une autre zone d'intérêt. En effet, l'oeil est seulement sensible aux détails dans le centre de son champ visuel, c'est pour cela que l'information visuelle est traitée seulement pendant les fixations et pas durant les saccades [Ray98, Duc03, War05]. Les saccades en combinaison avec les fixations aident à se former un modèle mental de la scène visuelle. Comme Guéhéneuc et Yusuf [Gué06, YKM07], nous utilisons les fixations comme indice de l'attention qu'un sujet porte sur les différents éléments d'un diagramme.

Nous utilisons aussi, une chaise de dentiste pour configurer l'environnement des expériences facilement (par exemple aligner la tête du sujet pour qu'elle soit au centre des capteurs de l'écran de l'ordinateur des expériences) sans mettre les sujets dans une position inconfortable et en même temps pour éviter que les sujets bougent pendant l'expérience. Nous avons aussi utilisé un oreiller de voyage pour donner plus de support au cou et diminuer les mouvements involontaires de la tête. L'écran utilisé pour visualiser les diagrammes de l'expérience, a une taille de 17 pouces et il est de type CRT. La figure 3.6 montre une photo de l'équipement utilisé pour faire l'expérience.

3.9 Procédure

Les sujets s'inscrivent pour l'expérience soit en personne, soit par courriel, soit à travers l'inscription en ligne sur le site Web des expériences (<http://www-etud.iro.umontreal.ca/~cepedapg/experience>). Le site Web donne une description du déroulement de l'expérience et un vidéo présentant l'oculomètre utilisé.

Les expériences ont lieu dans un endroit calme sans distraction. La procédure de l'expérience suit les étapes suivantes :

- la vérification de l'identité du sujet ;

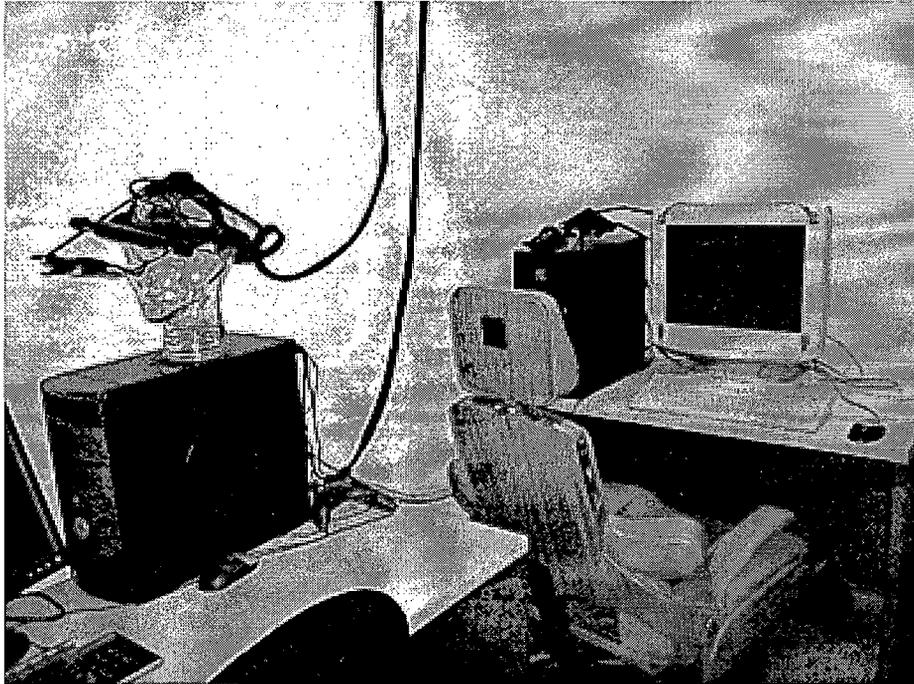


FIG. 3.6 – Equipement utilisé pour faire l'expérience

- la présentation de l'expérience, le survol de la procédure et de la durée estimée de l'expérience (qui dure environ 30-40 min) ;
- la mention du caractère confidentiel et anonyme de l'expérience ainsi que le droit du sujet d'arrêter l'expérience à n'importe quel moment sans conséquence ;
- la présentation d'un tutoriel sur les 4 représentations choisies pour l'expérience. Le tutoriel rappelle aussi aux sujets des concepts de patrons de conception et en même temps familiarise (sans être explicite) les sujets avec les tâches qu'ils devront faire plus tard. L'utilisation du tutoriel sert à donner aux sujets des connaissances de base et à diminuer l'anxiété qu'ils peuvent éprouver pendant l'exécution de l'expérience ;
- la présentation de l'oculomètre utilisé pour l'expérience ainsi que des instructions du mode d'emploi de l'équipement (par exemple : essayer de ne pas bouger la tête) ;

- l'installation du sujet sur la chaise, la pose de l'oreiller de voyage et le rappel de ne pas bouger pendant l'expérience pour avoir une meilleure qualité des données collectées ;
- les explications sur l'utilisation du clavier : la seule touche activée pour l'expérience est ESC et elle sert à changer de stimulus une fois chaque tâche finie (la souris n'est pas activée pour l'expérience) ;
- le lancement de l'application des expériences ;
- la calibration de l'oculomètre : la calibration est faite par le sujet en fixant son regard sur 9 points de l'écran qui sont montrés un par un de façon séquentielle ;
- la présentation du cadrice JHotDraw (sert à mettre en contexte, montrer les caractéristiques principales du cadrice et donner un aperçu du système) ;
- la collecte de données. Pendant la collecte de données le sujet fait 6 tâches : 2 tâches de composition de patrons, 2 tâches d'identification de rôles de classes et 2 tâches de participation de patrons. Pour un aperçu de la distribution de questions voir la figure 3.2. Il n'y a pas de limite de temps pour l'exécution de chaque tâche, cependant, nous avons demandé aux sujets de donner leur réponse à haute voix dès qu'ils ont la réponse. Nous avons utilisé une liste de contrôle pour vérifier les réponses de chaque tâche. Si tous les éléments de la liste sont corrects la réponse du sujet est correcte, autrement la réponse est considérée comme incorrecte ;
- l'administration d'un questionnaire d'auto-évaluation des connaissances des patrons de conception et du niveau de familiarité avec JHotDraw. Nous avons donné le questionnaire à la fin de l'expérience pour ne pas révéler le but de l'expérience.

À la fin de l'expérience, nous avons donné un cadeau symbolique aux sujets pour les remercier de leur participation. Nous leur avons aussi demandé de ne transmettre aucune information sur l'expérience avant la date de terminaison des expériences.

CHAPITRE 4

ANALYSE ET RÉSULTATS

À partir des données collectées pour l'expérience et après avoir vérifié la normalité des données des variables dépendantes, nous discutons de l'analyse des données et des résultats trouvés à partir de la vérification de nos hypothèses avec l'utilisation du test statistique « Student t -test ». Le t -test est un test statistique utilisé pour identifier la signification statistique du chevauchement de deux ensembles de données à partir de petits échantillons de données [Duc03]. La convention générale pour l'élimination des observations aberrantes (outliers) a été d'éliminer (après avoir vérifié l'origine possible de cette valeur aberrante sur les notes de chaque expérience et jugé s'il est pertinent de l'exclure) seulement les observations aberrantes lointaines, c'est-à-dire, celles supérieures à 3 quartiles de la médiane afin d'analyser le plus de données possibles. Le tableau 4.1 récapitule quelques informations d'intérêt sur les données collectées.

4.1 Analyse de données pour les diagrammes de 15 classes

Les tableaux 4.2, 4.3 et 4.4 résument l'effet des représentations D, G, S et U sur des diagrammes de classes UML pour les tâches P, C et R respectivement. Les valeurs de PRC, DMF, RFOA et RTFOA sont données pour chaque représentation. Les colonnes des « p -values » montrent la signification statistique des différences entre U_{15} et S_{15} (H_{0_1}), U_{15} et G_{15} (H_{0_2}) et U_{15} et D_{15} (H_{0_3}) respectivement. L'analyse de chacun de ces tableaux est abordée dans les sections suivantes.

Données collectées	
Nombre de sujets (#)	24
Données (Mb)	36
Nombre de vidéos (#)	144
Temps total d'expérimentations (heures)	18
Temps total oculomètre (min)	112,4
Nombre total de fixations	21.395
Diagrammes de 15 classes	
Temps moyen pour la tâche de participation de classes (C)(sec)	36
Temps moyen pour la tâche de composition de patrons (P)(sec)	33
Temps moyen pour la tâche d'indentification de rôles (R)(sec)	54
Diagrammes de 40 classes	
Temps moyen pour la tâche de participation de classes (C)(sec)	49
Temps moyen pour la tâche de composition de patrons (P)(sec)	52
Temps moyen pour la tâche d'indentification de rôles (R)(sec)	58

TAB. 4.1 – Données collectées

4.1.1 Analyse de données pour la tâche de participation de classes : P

Le tableau 4.2 résume l'effet de l'utilisation des représentations D, G, S et U sur les diagrammes de classes UML pour l'exécution de la tâche de participation de classes. Nous observons une meilleure performance dans U. Le pourcentage de réponses correctes est inférieur pour toutes les représentations D (PCR = 50%), G (PCR = 50%) et S (PCR = 83,33%) par rapport à U (PCR = 100%). Nous observons avec les valeurs de DMF que la moyenne pour le temps en U (DMF = 264,6 ms) est légèrement inférieure (et donc la charge cognitive pour interpréter la représentation est inférieure) aux autres représentations D (DMF = 277,6 ms), G (DMF = 270,6 ms), S (DMF = 273,8 ms). Cependant, il n'y a pas d'impact significatif des représentations sur cette métrique.

Les métriques du RFOA et du RTFOA donnent des résultats intéressants où nous observons des différences significatives dans l'effort cognitif entre D et U, les

ratios pour D (RFOA = 0,54 et RTFOA = 0,54) étant inférieurs à ceux de U (RFOA = 0,79 et RTFOA = 0,84). La figure 4.1 montre ces différences avec les médianes des ratios visiblement inférieures pour D (RFOA = 0,54 et RTFOA = 0,56) par rapport à U (RFOA = 0,8 et RTFOA = 0,86) et supérieures pour D (DMF = 266,28 ms) et U (DMF = 240,81 ms). Dans toutes les métriques, nous trouvons une variance plus petite pour D par rapport aux autres représentations. D est une représentation fortement textuelle et qui donne plus d'information que les autres représentations. Aussi, une variance plus petite pourrait venir du niveau de connaissance de patrons de conception et/ou du niveau de connaissance de JHot-Draw lors de l'exécution des tâches.

Pour les hypothèses étudiées, nous pouvons conclure que, pour la tâche de participation de classes, l'effort cognitif moyen des sujets pour trouver les classes pertinentes en utilisant D est supérieur à celui de U et aussi que l'efficacité de U (PRC = 100%) est supérieure à celle de D (PRC = 50%). Pour les autres deux représentations nous n'avons pas trouvé des différences significatives dans l'effort cognitif et le temps. Cependant, la représentation de notations de collaboration UML a aussi obtenu des meilleurs résultats par rapport à Gamma et Schauer pour la métrique de PRC. Il est important de mentionner que la représentation de Schauer a obtenue d'aussi bons résultats que les diagrammes de collaboration UML pour la charge cognitive et le temps.

	P_D15	P_G15	P_S15	P_U15	p-value (H_{01})	p-value (H_{02})	p-value (H_{03})
PRC (%)	50	50	83,33	100	—	—	—
DMF (ms)	277,61	270,60	273,85	264,66	0,79	0,90	0,7
RFOA (#)	0,54	0,70	0,84	0,79	0,29	0,12	0,00
RTFOA (ms)	0,55	0,74	0,87	0,84	0,38	0,08	<0,0001

TAB. 4.2 – Effet des représentations D, G, S et U pour la tâche P, diagramme de 15 classes

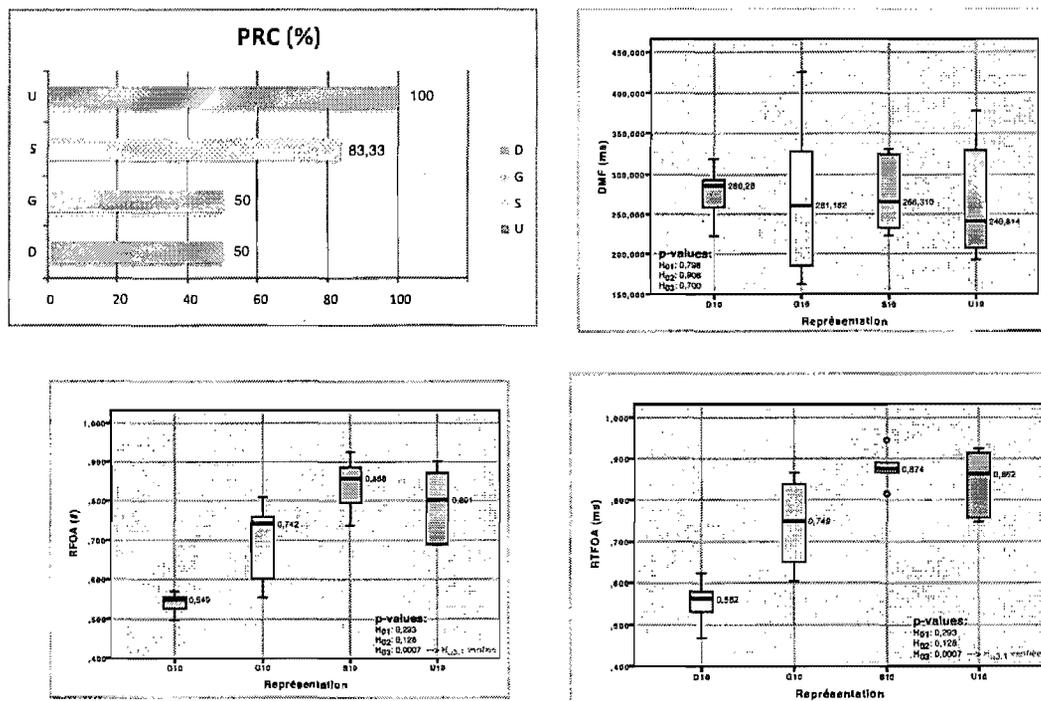


FIG. 4.1 – Distribution de données pour P, diagramme de 15 classes

4.1.2 Analyse de données pour la tâche de composition de patrons : C

Pour la tâche de composition de patrons les valeurs de PCR montrent une meilleure performance de D et S (PCR = 100%) par rapport à U (PCR = 83,33%). Les valeurs de DMF montrent des différences dans le temps mais sans donner de différences significatives. À partir des métriques RFOA et RTFOA, nous trouvons des différences significatives entre D et U et G et U respectivement, montrant une difficulté supérieure pour trouver les classes pertinents à la tâche dans U par rapport à D et G. La figure 4.2 montre plus en détail ces différences. Nous pouvons voir les différences entre les médianes pour D (RFOA = 0,65 et RTFOA = 0,73), G (RFOA = 0,66 et RTFOA = 0,71) et U (RFOA = 0,45 et RTFOA = 0,51).

Les valeurs pour D et G sont très similaires. Cela peut s'expliquer par la disposition des informations dans les représentations (dans D et G l'information relative

à C est située dans la même zone, tandis que pour U l'information est située dans des zones différentes).

	P_D15	P_G15	P_S15	P_U15	p-value (H_{01})	p-value (H_{02})	p-value (H_{03})
PRC (%)	100	50	100	83,33	—	—	—
DMF (ms)	280,11	279,76	230,84	245,12	0,56	0,42	0,33
RFOA (#)	0,66	0,65	0,56	0,42	0,21	0,03	0,01
RTFOA (ms)	0,73	0,70	0,63	0,45	0,09	0,02	0,00

TAB. 4.3 – Effet des représentations D, G, S et U pour la tâche C, diagramme de 15 classes

Pour les hypothèses étudiées, nous pouvons conclure que, pour la tâche de composition de patrons, l'effort cognitif moyen des sujets pour trouver les classes pertinentes en utilisant D et G est inférieur à celui de U. Cependant, à partir de la métrique de PRC nous pouvons seulement dire que D est plus efficace que U, mais pas forcément G (tableau 4.3).

4.1.3 Analyse de données pour la tâche d'identification de rôles : R

Pour cette tâche les valeurs de PCR montrent une meilleure performance de D (PCR = 100%) par rapport à U (PCR = 50%). Les valeurs de DMF ne montrent pas de différences significatives dans le temps pour chaque représentation. À partir des métriques RFOA et RTFOA, nous trouvons des différences significatives entre D et U et S et U respectivement, montrant cette fois-ci une difficulté supérieure pour trouver les classes pertinentes à la tâche dans U par rapport à D et S. La figure 4.3 montre les différences entre les médianes pour D (RFOA = 0,76 et RTFOA = 0,81), S (RFOA = 0,85 et RTFOA = 0,88) et U (RFOA = 0,52 et RTFOA = 0,63).

Pour les hypothèses étudiées, nous pouvons conclure que, pour la tâche d'identification de rôles, l'effort cognitif moyen des sujets pour trouver les classes perti-

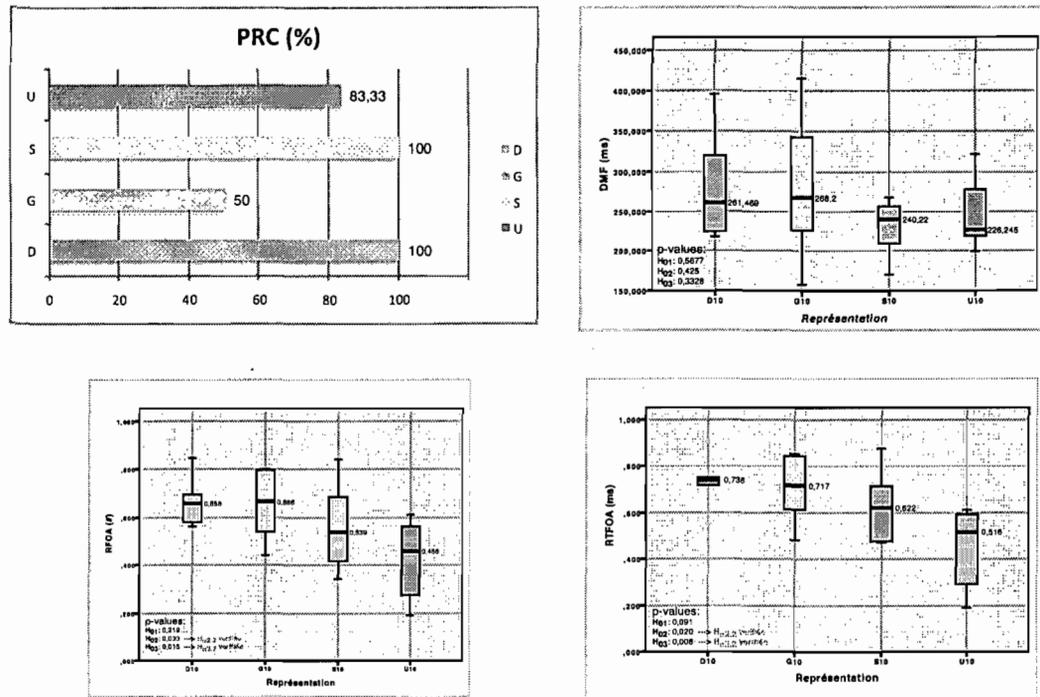


FIG. 4.2 – Distribution de données pour C, diagramme de 15 classes

nentes en utilisant D et S est inférieur à celui de U. Cela peut s'expliquer pour la disposition des informations dans D. Cependant, à partir de la métrique de PRC, nous pouvons seulement dire que D est plus efficace que U, mais pas forcément S. La valeur supérieure des métriques de RFOA et RTFOA peut indiquer en fait une bonne identification des classes pertinentes de la part des sujets. Cependant, nous observons aussi, un effort cognitif supérieur ($DMF = 289,36$ ms) causé par le manque d'information dans la représentation, cela peut expliquer pourquoi aucun sujet n'a répondu correctement à cette tâche (voir tableau 4.4).

4.1.4 Analyse de l'impact des variables d'atténuation

Les résultats présentés dans les sections précédentes montrent une meilleure performance de U par rapport à D pour la tâche P, ainsi qu'une meilleure performance de D par rapport à U pour les tâches C et R.

	P_D15	P_G15	P_S15	P_U15	p-value (H_{0_1})	p-value (H_{0_2})	p-value (H_{0_3})
PRC (%)	100	40	0	50	—	—	—
DMF (ms)	327,06	268,43	289,36	272,85	0,57	0,91	0,14
RFOA (#)	0,73	0,54	0,87	0,52	0,00	0,57	0,01
RTFOA (ms)	0,79	0,60	0,87	0,58	0,00	0,81	0,02

TAB. 4.4 – Effet des représentations D, G, S et U pour la tâche R, diagramme de 15 classes

Pour la tâche C l'effort cognitif pour trouver les classes pertinentes avec G est inférieur à celui de U. Cependant, la valeur de PRC est inférieure à celle de U, nous empêchant de tirer des conclusions plus intéressantes. En considérant les variances entre DMF, RFOA et RTFOA, déjà mentionnées dans la section 4.1.1, (voir par exemple la différence entre les variances de G par rapport à D pour la tâche C), nous avons décidé d'étudier si le niveau de connaissance de patrons de conception pourrait atténuer les résultats obtenus. Nous avons éliminé la variable de niveau de connaissance de JHotDraw de cette étude, car plus de 80% de participants avaient un niveau de connaissance basique du cadriceil et donc nous sommes certains que la variable n'aura pas d'impact dans notre étude. Nous avons décidé de grouper les niveaux de connaissance de patrons de conception en 2 catégories : 1 pour le niveau basique et intermédiaire et 2 pour le niveau expert. Pour des raisons de simplicité, nous avons choisi d'étudier les valeurs des réponses données pour les sujets ainsi que les valeurs de RFOA.

Le tableau 4.5 donne les valeurs de signifiante du test 2-way ANOVA pour l'impact du niveau de connaissance de patrons de conception et l'impact combiné (Représentation \times Niveau de connaissance de patrons de conception) pour les tâches C, P et R. Nous trouvons un résultat intéressant pour le niveau de connaissance de patrons de conception et son impact sur les réponses pour la tâche C. La valeur du p -value égale à 0,028, indique qu'il y a une différence significative dans la performance entre sujets avec différents niveaux de connaissance de patrons de

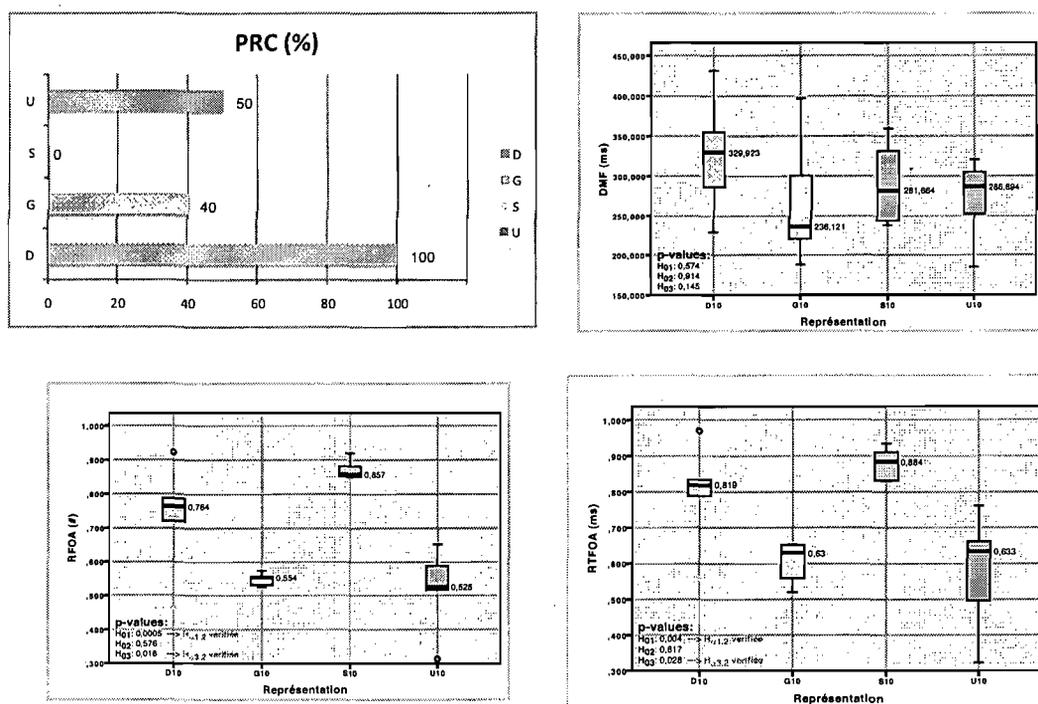


FIG. 4.3 – Distribution de données pour R, diagramme de 15 classes

conception et donc nous pouvons considérer la possibilité qu'effectivement G soit plus efficace que U pour la tâche C quand les sujets ont un très bonne expertise en patrons de conception. La figure 4.4 montre cet impact sur les représentations G et U (premier diagramme à gauche en haut) où pour G il existe une grande différence entre les réponses des sujets avec niveaux 1 et 2.

	Réponses (C)	RFOA (C)	Réponses (P)	RFOA (P)	Réponses (R)	RFOA (R)
Niv_Connaissance_DP	0,02	0,31	0,82	0,21	0,45	0,95
Combiné	0,13	0,95	0,98	0,93	0,15	0,06

TAB. 4.5 – Impact du niveau de connaissance de patrons de conception, diagramme de 15 classes

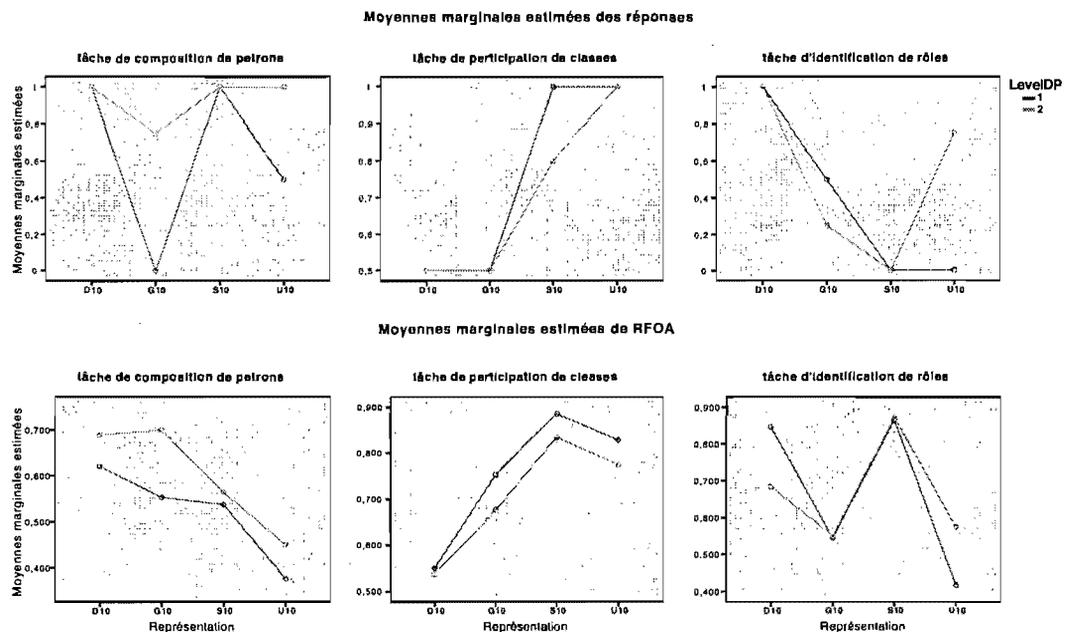


FIG. 4.4 – Impact combiné de connaissance de patrons de conception et représentations, diagramme de 15 classes

4.2 Analyse de données pour les diagrammes de 40 classes

Les tableaux 4.6, 4.7 et 4.8 résument l'effet des représentations D, G, S et U sur des diagrammes de classes UML pour les tâches P, C et R respectivement. Les valeurs de PRC, DMF, RFOA et RTFOA sont données pour chaque représentation. Les colonnes des « p -values » montrent la signification statistique des différences entre U40 et S40 (H_{0_1}), U40 et G40 (H_{0_2}) et U40 et D40 (H_{0_3}) respectivement. Il faut remarquer que pour cette analyse, la lisibilité des diagrammes est réduite (nous aborderons ce sujet plus en profondeur dans la section sur les menaces à la validité). L'analyse de chacun de ces tableaux est abordée dans les sections suivantes.

4.2.1 Analyse de données pour la tâche de participation de classes : P

Pour P, nous trouvons comme dans les diagrammes de 15 classes, une efficacité inférieure de D par rapport à U avec un taux de PRC de 50% contre 83,33% res-

pectivement. Les valeurs de RFOA ($D= 0,37$ $U= 0,61$) et RTFOA ($D=0,4$, $U= 0,68$) montrent aussi des différences significatives dans l'effort cognitif pour trouver les classes pertinentes entre D et U. Les valeurs très similaires pour DMF n'ont donné aucun résultat significatif (tableau 4.6).

	P_D15	P_G15	P_S15	P_U15	p-value (H_{0_1})	p-value (H_{0_2})	p-value (H_{0_3})
PRC (%)	50	66,66	83,33	83,33	—	—	—
DMF (ms)	292,10	277,40	324,74	331,72	0,92	0,38	0,52
RFOA (#)	0,37	0,27	0,43	0,61	0,29	0,21	0,01
RTFOA (ms)	0,4	0,57	0,80	0,68	0,13	0,14	0,00

TAB. 4.6 – Effet des représentations D, G, S et U pour la tâche P, diagramme de 40 classes

La figure 4.4 montre clairement un effort cognitif plus grand avec l'utilisation de D par rapport à U. Il est aussi intéressant de remarquer ici la bonne performance de S (PRC = 83,3%, RFOA= 0,43 et RTFOA = 0,80) qui est très similaire à celle de U. Pour cette tâche, nous pouvons conclure que l'effort cognitif pour trouver les classes pertinentes utilisant D est supérieur à celui de U et aussi que l'efficacité de U est supérieure à D (données obtenues à partir de PCR).

4.2.2 Analyse de données pour la tâche de composition de patrons : C

Le tableau 4.7 résume les valeurs obtenues pour les 4 représentations. Pour la métrique de PRC, U (PRC = 100%) montre une meilleure performance par rapport aux autres représentations D (PRC = 66,6%), G (PRC = 50%) et S (PRC = 83,33%). Dans la figure 4.6, nous pouvons observer, comme pour les diagrammes de 15 classes, un effort cognitif inférieur pour trouver les classes pertinentes dans D pour les métriques de RFOA et RTFOA par rapport à U mais aussi un effort cognitif supérieur dans D pour interpréter la représentation. Cela peut être causé par le manque d'éléments graphiques dans la représentation D et la lisibilité réduite

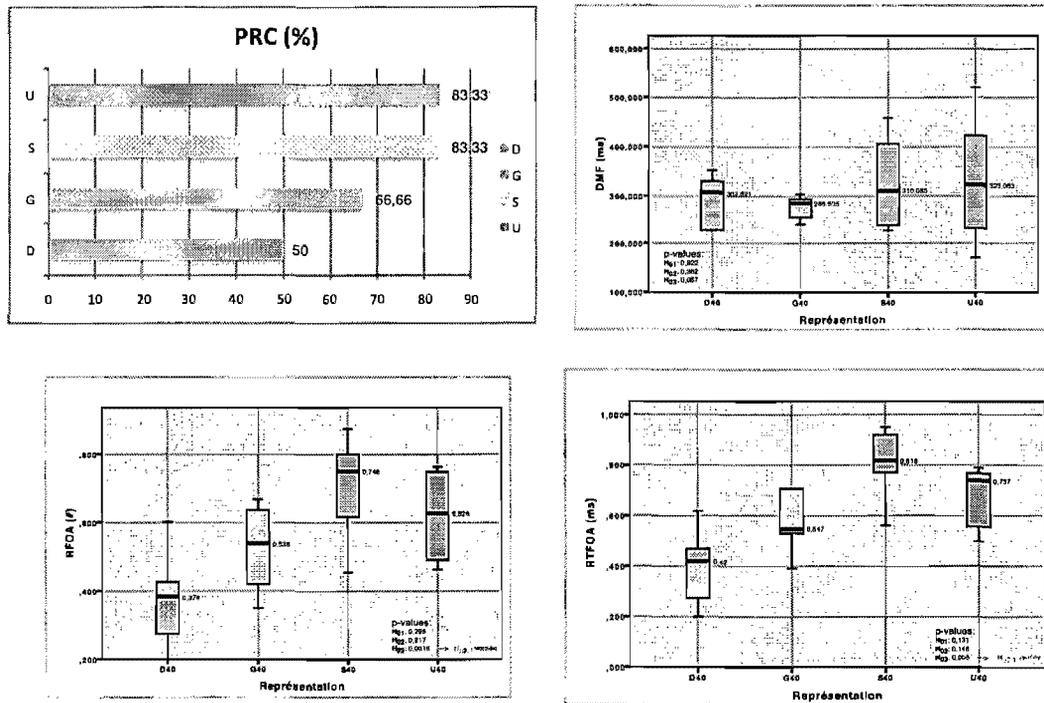


FIG. 4.5 – Distribution de données pour P, diagramme de 40 classes

pour la grande densité de classes dans la représentation. Cependant les tests de signifiante dans le tableau 4.7 ne montrent pas de différences significatives entre les représentations.

	P_D15	P_G15	P_S15	P_U15	p-value (H_{01})	p-value (H_{02})	p-value (H_{03})
PRC (%)	66,66	50	83,33	100	—	—	—
DMF (ms)	328,53	264,10	266,81	246,52	0,52	0,47	0,05
RFOA (#)	0,56	0,27	0,43	0,40	0,77	0,31	0,31
RTFOA (ms)	0,62	0,30	0,51	0,41	0,48	0,41	0,24

TAB. 4.7 – Effet des représentations D, G, S et U pour la tâche C, diagramme de 40 classes

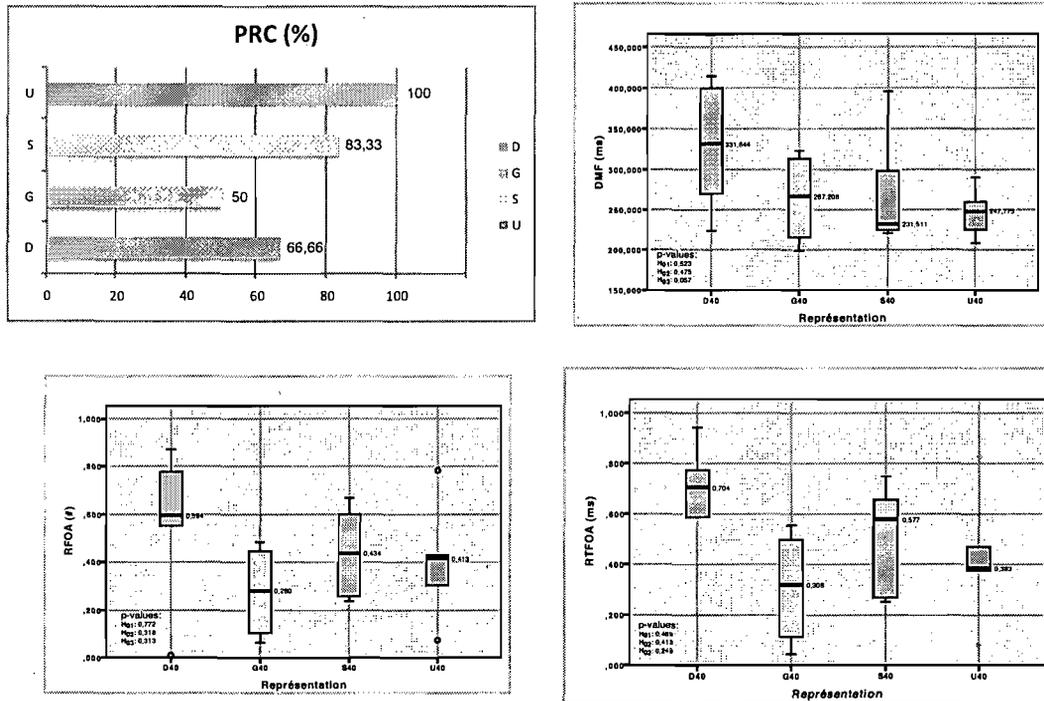


FIG. 4.6 – Distribution de données pour C, diagramme de 40 classes

4.2.3 Analyse de données pour la tâche d'identification de rôles : R

Les valeurs de PCR pour cette tâche montrent des performances similaires pour G et U (PRC = 83,33 %) et légèrement inférieures pour D (PRC = 66,66 %). La valeur de PRC égale à 0% pour S est un nouvel indice du manque d'information dans la représentation. Les valeurs de DMF montrent encore un effort cognitif supérieur dans D (DMF = 303,9) pour interpréter la représentation mais sans être une différence significative par rapport aux autres représentations G (DMF = 295,5 ms), S (DMF = 286,3 ms) et U (DMF = 294,9 ms). Pour les valeurs de RFOA et RTFOA, nous observons une légère différence entre l'effort cognitif pour trouver les classes pertinentes en D et S par rapport à U. En conclusion, à partir des données présentées dans le tableau 4.8, nous pouvons dire qu'il n'y a pas de différence significative dans l'effort cognitif entre les représentations pour la tâche R.

	P_D15	P_G15	P_S15	P_U15	p-value (H_{01})	p-value (H_{02})	p-value (H_{03})
PRC (%)	66,66	83,33	0	83,33	—	—	—
DMF (ms)	303,94	295,59	286,33	294,9	0,82	0,98	0,84
RFOA (#)	0,57	0,50	0,70	0,41	0,03	0,49	0,28
RTFOA (ms)	0,63	0,54	0,72	0,48	0,12	0,69	0,32

TAB. 4.8 – Effet des représentations D, G, S et U pour la tâche R, diagramme de 40 classes

4.2.4 Analyse de l'impact des variables d'atténuation

Les résultats présentés dans les sections précédentes montrent une meilleure performance de U par rapport à D pour la tâche P, mais aucune différence entre les représentations pour les autres tâches. Comme pour la section 4.1, nous avons décidé d'étudier si le niveau de connaissance de patrons de conception pourrait atténuer les résultats obtenus. Les variables dépendantes choisies sont à nouveau les réponses des sujets et RFOA.

Le tableau 4.9 donne les valeurs de signifiante du test 2-way ANOVA pour l'impact du niveau de connaissance de patrons de conception et l'impact combiné (Représentation \times Niveau de connaissance de patrons de conception) pour les tâches C, P et R. Nous ne trouvons de différences significatives pour aucune variable dans aucune tâche. Cependant, les p -values de 0,056 pour l'impact du niveau de connaissance de patrons de conception sur les réponses (C) et de 0,057 et 0,058 pour l'impact du niveau de connaissance de patrons de conception et le combiné sur RFOA (C) pourraient indiquer un léger avantage pour les représentations avec plus d'éléments graphiques par rapport à celles plus textuelles (voir le taux de PRC dans la figure 4.6 et les moyennes marginales estimées des réponses pour C dans la figure 4.8) à cause de la lisibilité réduite pour les diagrammes de 40 classes. En plus, ces valeurs pourraient indiquer le besoin de l'exécution d'une étude de réplication.

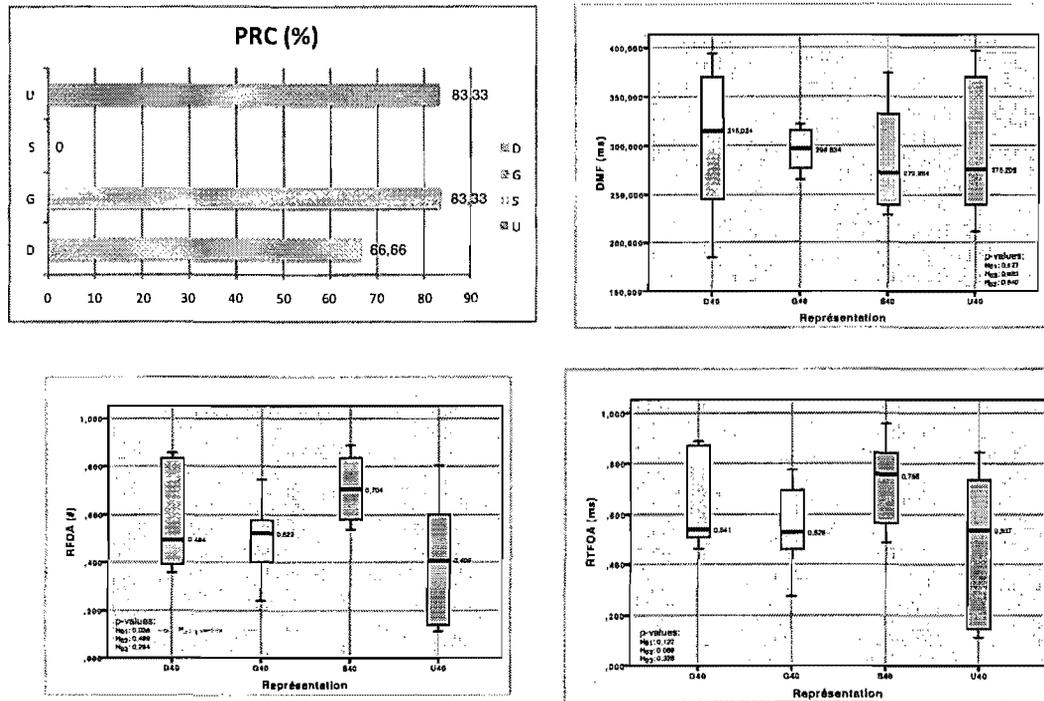


FIG. 4.7 – Distribution de données pour R, diagramme de 40 classes

4.3 Conclusion

Les analyses ont montré que pour les diagrammes de 15 classes : pour la tâche de participation de classes (P) les hypothèses suivantes ont été vérifiées :

- H_{P0_1} : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations U et S. Cependant, même si la valeur de PCR est légèrement supérieure pour U (PCR en U = 100%, PCR en S = 83,33%), il faut remarquer que S a obtenu d'aussi bons résultats que U pour la charge cognitive et le temps.
- H_{P0_2} : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations G et S. La valeur de PCR en U est supérieure de 50% à celle de G.
- $H_{P\alpha_{3.1}}$: l'effort cognitif moyen des sujets face à D est supérieur quand on le compare avec U. La valeur de PCR en U est aussi supérieure de 50% à celle

	Réponses (C)	RFOA (C)	Réponses (P)	RFOA (P)	Réponses (R)	RFOA (R)
Niv_Connaissance_DP	0,05	0,88	0,22	0,05	0,42	0,20
Combiné	0,63	0,65	0,34	0,05	0,18	0,15

TAB. 4.9 – Impact du niveau de connaissance de patrons de conception, diagramme de 40 classes

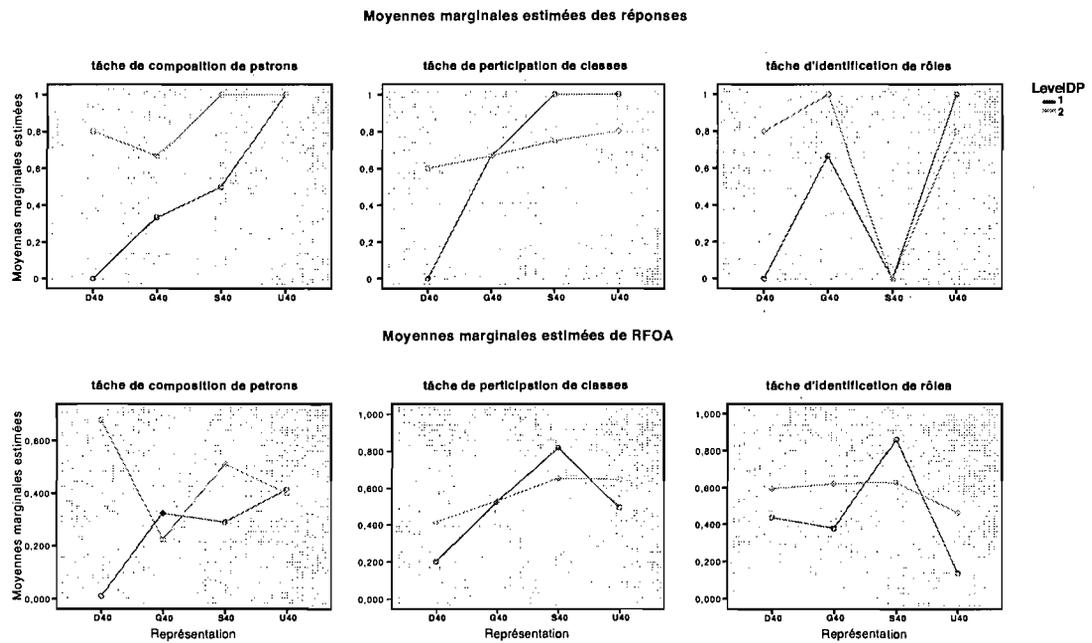


FIG. 4.8 – Impact combiné de connaissance de patrons de conception et représentations, diagramme de 40 classes

de D.

Pour la tâche de composition de classes (C) les hypothèses suivantes ont été vérifiées :

- H_{C0_1} : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations U et S. La valeur de PCR est légèrement supérieure pour S (PCR en U = 83,33%, PCR en S = 100%).
- $H_{C\alpha_2.2}$: l'effort cognitif moyen des sujets face à G est inférieur quand on le compare avec U. Cependant, la valeur de PCR en U est supérieure de 50% à celle de G. Une étude postérieure sur l'influence du niveau de connaissance

des patrons de conception sur la performance des sujets pour la tâche C a montré que, pour G, le niveau de connaissance de patrons de conception a une influence significative dans la performance des sujets. Nous pouvons donc considérer la possibilité qu'effectivement G soit plus efficace que U pour la tâche C quand les sujets ont une très bonne expertise en patrons de conception.

- $H_C\alpha_{3.2}$: l'effort cognitif moyen des sujets face à D est inférieur quand on le compare avec U. La valeur de PCR en D est aussi légèrement supérieure à celle de U.

Pour la tâche d'identification de rôles (R) les hypothèses suivantes ont été vérifiées :

- $H_R\alpha_{1.2}$: l'effort cognitif moyen des sujets face à S est inférieur quand on le compare avec U. Cependant, même s'il n'est pas significatif, on trouve un effort cognitif supérieur pour interpréter S. La valeur de PCR pour S est de 0% à cause, peut-être, du manque d'information dans la représentation.
- H_R0_2 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations G et S. Les valeurs de PCR en G et U sont elles aussi très similaires.
- $H_R\alpha_{3.2}$: l'effort cognitif moyen des sujets face à D est inférieur quand on le compare avec U. La valeur de PCR en D est supérieure à celle de U de 50%.

Les analyses ont montré que pour les diagrammes de 40 classes : pour la tâche de participation de classes (P) les hypothèses suivantes ont été vérifiées :

- H_P0_1 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations U et S. Même la valeur de PCR est égale pour U et S (83,33%). S a obtenu d'aussi bons résultats que U dans l'effort cognitive et le temps.
- H_P0_2 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations G et S. La valeur de PCR en U est aussi supérieure à celle de G.
- $H_P\alpha_{3.1}$: l'effort cognitif moyen des sujets face à D est supérieur quand on le

compare avec U. La valeur de PCR en U est aussi supérieure à celle de D. Pour la tâche de composition de classes (C) les hypothèses suivantes ont été vérifiées :

- H_C0_1 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations U et S. La valeur de PCR est légèrement supérieure pour U (PCR en U = 100%, PCR en S = 83,33%).
- H_C0_2 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations U et G. La valeur de PCR en U est supérieure de 50% à celle de G.
- H_C0_3 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations U et D. La valeur de PCR en D est aussi inférieure à celle de U.

Pour la tâche d'identification de rôles (R), les hypothèses suivantes ont été vérifiées :

- H_R0_1 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations S et U. La valeur de PCR pour S est de 0% à cause, peut-être, du manque d'information dans la représentation.
- H_R0_2 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations G et U. Les valeurs de PCR en G et U sont égales.
- H_R0_3 : il n'y a pas de différence dans l'effort cognitif moyen des sujets entre les représentations G et U. La valeur de PCR en D est inférieure à celle de U.

Nous avons aussi rapporté que pour les diagrammes de 40 classes, la lisibilité est réduite par la grande densité de classes, ce qui peut influencer les résultats obtenus.

CHAPITRE 5

MENACES À LA VALIDITÉ

À partir de [WRH⁺00], nous avons identifié des menaces à la validité de notre expérience et, dans la mesure du possible, nous les avons traitées ou acceptées selon le cas. Quelques menaces à la validité sont reliées à l'utilisation de sujets humains dans l'étude, d'autres à l'utilisation de l'équipement.

5.1 Validité interne

Les menaces à la validité interne sont des influences qui peuvent affecter les valeurs des variables indépendantes en ce qui concerne la causalité [WRH⁺00]. Nous avons identifié trois possibles menaces à la validité interne : maturation, instrumentation et diffusion des traitements.

Pour la menace de la maturation, nous avons traité l'effet d'apprentissage en présentant les tâches aux sujets dans un ordre aléatoire et différent, ce qui évite la probabilité de favoriser les tâches présentées à la fin de l'expérience ou défavoriser ces mêmes tâches à cause de la fatigue. Nous avons aussi essayé de diminuer dans la mesure du possible l'effet de fatigue avec la conception de notre expérience (6 questions d'environ 1-2 minutes pour trouver la réponse, le temps maximum pour répondre à tous les questions est d'environ 15 minutes par sujet).

La menace de l'instrumentation est liée à l'utilisation de l'équipement. Les sujets doivent minimiser les mouvements de la tête pour éviter de décalibrer l'oculomètre et le bandeau de celui-ci est légèrement pesant. Les mouvements de la tête sont inévitables, car pendant que les sujets sont concentrés à donner la réponse, ils oublient souvent la restriction du mouvement de la tête. Pour minimiser cette menace,

nous avons utilisé une chaise de dentiste et un oreiller de voyage pour donner plus de support au cou et diminuer les mouvements involontaires de la tête. Dans quelques cas, nous avons trouvé un léger décalage des coordonnées des fixations des sujets. Pour corriger ce décalage, nous avons eu recours à des corrections de la dérive sur les données. Nous avons aussi identifié et accepté une menace à l'instrumentation causée par la résolution maximale possible pour l'affichage des diagrammes (1024 × 768 pixels) dans notre système, cela produit une diminution en la lisibilité des diagrammes de 40 classes.

Finalement, pour éviter la diffusion des informations de l'expérience et donc empêcher des sujets d'apprendre les traitements d'avance, nous avons indiqué à chaque sujet de ne pas parler de l'expérience avant la date de la fin des expériences. Nous croyons que les sujets ont suivi ces instructions.

5.2 Validité de construction

La menace à la validité de construction consiste au fait de généraliser le résultat de l'expérience au concept ou à la théorie sur lequel s'appuie l'expérience [WRH⁺00]. Nous avons identifié et traité trois menaces à la validité de construction : le biais mono-méthode, l'estimation d'hypothèse et l'appréhension. Nous avons identifié et accepté la menace du biais mono-opération.

Le biais mono-méthode est le risque d'avoir un biais dans les mesures de notre expérience qui peut être causé par l'utilisation d'un seul type de mesure. Pour traiter le biais mono-méthode nous avons décidé d'utiliser quatre variables dépendantes : PRC, DMF, RFOA et RTFOA et de faire une vérification croisée des résultats entre les variables pour l'analyse des résultats.

Nous avons traité la menace d'estimation des hypothèses en n'informant pas

les sujets du but de l'expérience. Les sujets ont été seulement informés pendant le tutoriel qu'ils devaient faire des tâches sur différentes représentations de patrons de conception.

Pour éviter l'appréhension, nous avons en premier expliqué aux sujets le fonctionnement de l'occulomètre et nous les avons rassurés sur l'absence de risques liés à l'exposition des yeux aux rayons infra-rouges. Deuxièmement, nous avons rassuré les sujets sur l'anonymat de leurs données et de leurs réponses. Finalement, nous n'avons pas défini un temps limite pour l'exécution de tâches, nous les avons seulement indiqués de prendre le temps nécessaire pour donner leurs réponses.

Nous acceptons l'existence d'un biais de mono-opération causé pour l'utilisation d'un seul système (JHotDraw) pour notre expérience. Cependant, vue la quantité de variables à évaluer, nous avons décidé qu'il valait mieux accepter ce risque qu'augmenter la complexité de la conception de notre expérience en y ajoutant plus de variables.

5.3 Validité externe

Les menaces à la validité externe sont des conditions qui limitent la généralisation des résultats de notre expérience [WRH⁺00]. Nous avons traité deux types d'interactions avec les traitements : l'interaction de sélection et l'interaction du réglage.

Pour l'interaction de sélection, le problème est de savoir si les sujets sont ou ne sont pas représentatifs des professionnels en génie logiciel. Dans notre expérimentation, tous les sujets sélectionnés pour faire l'expérience sont des étudiants de deuxième ou troisième cycle avec une bonne connaissance et expérience en UML et patrons de conception semblable à celle de la plupart des professionnels en génie logiciel.

Pour l'interaction du réglage, le menace est en relation avec la taille et com-

plexité des diagrammes utilisés. Pour traiter cette menace, nous avons choisi le logiciel JHotDraw qui est un bon exemple des logiciels qui utilisent amplement des patrons de conception. Les diagrammes présentés aux sujets contiennent quatre patrons de conception différents avec une densité de deux patrons de conception pour la composition. Nous avons affiché des diagrammes avec deux tailles différents. Les premiers diagrammes contiennent 15 classes, ce qui est le nombre commun de classes utilisées pour des activités de compréhension de logiciel [Amb05]. Les deuxièmes contiennent 40 classes pour étudier les effets des représentations dans des diagrammes plus complexes. 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.

5.4 Validité de conclusion

Pour les menaces à la validité de conclusion nous avons traité : la violation des hypothèses des tests statistiques, la fiabilité des mesures, les insignifiances aléatoires des réglages de l'expérience et l'hétérogénéité aléatoire des sujets.

Pour traiter la première menace, comme nous l'avons expliqué dans le chapitre précédent, nous avons vérifié et respecté les hypothèses des tests statistiques utilisés pour l'analyse de données. Pour traiter les menaces sur la fiabilité des mesures, nous avons choisi des mesures bien documentées dans la littérature et nous avons pris soin de faire une bonne calibration de l'oculomètre pour la collection des données. Pour traiter les insignifiances aléatoires des réglages de l'expérience, nous avons utilisé un laboratoire calme pour faire les expériences sans aucune distraction. Nous avons aussi fait des tests préliminaires d'exécution avant de commencer de façon officielle les expériences (voir chapitre 3).

À propos de la menace de l'hétérogénéité aléatoire des sujets, nous pensons que

notre échantillon est suffisamment hétérogène au niveau des connaissances pour refléter dans une grande partie la population cible. De plus, pour éviter que les résultats soient liés principalement à la connaissance des sujets, nous avons utilisé ce facteur comme une variable d'atténuation et nous avons vérifié l'impact de celui-ci (voir chapitre 4).

CHAPITRE 6

CONCLUSION

L'étude de l'efficacité des représentations visuelles de patrons de conception est une façon concrète d'améliorer la compréhension de logiciel. La représentation visuelle la plus utilisée pour décrire l'architecture d'un système est celle des diagrammes de classes UML et plusieurs travaux de recherche ont déjà été réalisés avec le but d'améliorer la compréhension de ce type de diagrammes. Actuellement le standard pour la représentation visuelle de patrons de conception est celui de la notation de collaboration UML.

Plusieurs auteurs ont signalé qu'il y a des lacunes dans cette représentation et ils ont proposé des alternatives avec le but d'améliorer la compréhension de patrons de conception. Cependant, à notre connaissance, aucune étude empirique n'avait été réalisée pour prouver ces améliorations. Dans notre étude, nous avons retenu 3 représentations à partir de l'analyse faite dans l'état de l'art.

Le but de notre 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 entre les représentations de Schauer et Keller (S) [SK98], Gamma et al. (G) [Vli98] et Dong et al. (D) [DYZ07] et la représentation standard (U) pendant l'exécution de tâches basiques à la compréhension de patrons de conception. Nous avons proposé de :

1. Comparer les performances des ingénieurs logiciels face à S [SK98] par rapport à U.
2. Comparer les performances des ingénieurs logiciels face à G [Vli98] par rapport à U.

3. Comparer les performances des ingénieurs logiciels face à D [DYZ07] par rapport à U.

Nous avons fait la conception et l'exécution d'une expérience pour collecter et comparer des données sur les performances des ingénieurs logiciels face à chaque représentation pendant qu'ils exécutaient les tâches de : participation de classes, composition de patrons de conception et identification des rôles joués par une classe. Les patrons de conception présentés dans les diagrammes sont : Composite, Prototype, Template Method, State et Singleton. Nous avons évalué chaque représentation dans des diagrammes de classes de petite densité (15 classes) et de grande densité (40 classes). Nous avons évalué séparément les diagrammes de 15 classes et les diagrammes de 40 classes, car la charge cognitive est différente. Les diagrammes montrent les patrons de conception avec une structure similaire à la représentation canonique telle que décrite dans [GHJV98] (l'implémentation des patrons de conception pour JHotDraw change légèrement la structure de la représentation canonique.)

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 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 notation de collaboration UML 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 dans un patron de conception indépendamment de la densité de classes.

En regardant la performance de S pour la tâche d'identification de rôles joués par un classe face à des modifications dans la structure du patron de conception, nous pouvons aussi remettre en question l'hypothèse qui suggère que montrer seulement les patrons de conception dans leur forme canonique est suffisant pour que les développeurs puissent comprendre l'implémentation du patron de conception. En effet, Vissilides [Vli98] a déjà souligné qu'il ne faut pas considérer les patrons de conception simplement par leur structure. Les patrons de conception sont juste des exemples et non des spécifications et leur structure peut très bien changer d'une implémentation à une autre. Cependant, il est nécessaire de faire d'autres expériences pour pouvoir confirmer les résultats obtenus et pouvoir généraliser pour tous les systèmes et tous les patrons de conception.

Comme travaux futurs, nous prévoyons d'abord faire :

- des études de réplication sur d'autres systèmes pour confirmer nos observations et traiter les différentes menaces à la validité que nous avons identifiées pour cette expérience ;
- des études plus axées sur la performance des sujets à partir de leur niveau de connaissance des patrons de conception pour trouver des patrons de lecture ;
- des études de réplication en utilisant d'autres patrons de conception ;
- proposer un nouvelle représentation à partir des résultats trouvés et faire des études pour tester son efficacité ;
- des études sur des méthodes de visualisation dynamique des informations relatives aux patrons de conception.

BIBLIOGRAPHIE

- [ACC+07] Lérina Aversano, Gerardo Canfora, Luigi Cerulo, Concettina Del Grosso, and Massimiliano Di Penta. An empirical study on the evolution of design patterns. In *ESEC-FSE '07 : Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pages 385–394, New York, NY, USA, 2007. ACM.
- [Amb05] Scott W. Ambler. *The Elements of UML(TM) 2.0 Style*. Cambridge University Press, New York, NY, USA, 2005.
- [Boj05] A. Bojko. Eye tracking in user experience testing : How to make the most of it. In *14th Annual Conference of the Usability Professionals Association (UPA)*, 2005.
- [BT06] Roman Bednarik and Markku Tukiainen. An eye-tracking methodology for characterizing program comprehension processes. In *ETRA '06 : Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 125–132, New York, NY, USA, 2006. ACM.
- [CK05] Christopher F. Chabris and Stephen M. Kosslyn. Representational correspondence as a basic principle of diagram design. In *Knowledge and Information Visualization*, pages 36–57, 2005.
- [Duc03] Andrew T. Duchowski. *Eye Tracking Methodology : Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
- [DYZ05] Jing Dong, Sheng Yang, and Kang Zhang. Visdp : A web service for visualizing design patterns on demand. *itcc*, 2 :385–391, 2005.
- [DYZ07] Jing Dong, Sheng Yang, and Kang Zhang. Visualizing design patterns in their applications and compositions. *IEEE Transactions on Software Engineering*, 33(7) :433–453, 2007.

- [Eic03] Holger Eichelberger. Nice class diagrams admit good design ? In *SoftVis '03 : Proceedings of the 2003 ACM symposium on Software visualization*, pages 159–ff. ACM, 2003.
- [EvG03] Holger Eichelberger and Jürgen Wolff von Gudenberg. Uml class diagrams - state of the art in layout techniques. In *Proceedings of Vissoft 2003, International Workshop on Visualizing Software for Understanding and Analysis*, pages 30–34, 2003.
- [Eye08] SR Research Eyelink. Sr research eyelink website. <http://www.eyelinkinfo.com/>, Fevrier 2008.
- [EYG97] A.H. Eden, A. Yehudai, and J. Gil. Precise specification and automatic application of design patterns. *International Conference on Automated Software Engineering (ASE '97)*, 0 :143–152, 1997.
- [FKGS04] Robert B. France, Dae-Kyoo Kim, Sudipto Ghosh, and Eunjee Song. A uml-based pattern specification technique. *IEEE Transactions on Software Engineering*, 30(3) :193–206, 2004.
- [GHJV98] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1998.
- [GK99] Joseph H. Goldberg and Xerxes P. Kotval. Computer interface evaluation using eye movements : methods and constructs. *International Journal of Industrial Ergonomics*, 24(6) :631–645, October 1999.
- [Gro01] Object Management Group. Unified modeling language specification, version 1.4. <http://www.omg.org>, 2001.
- [GSL⁺02] Joseph H. Goldberg, Mark J. Stimson, Marion Lewenstein, Neil Scott, and Anna M. Wichansky. Eye tracking in web search tasks : design implications. In *ETRA '02 : Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 51–58. ACM, 2002.
- [Gué06] Yann-Gaël Guéhéneuc. Taupe : towards understanding program comprehension. In *CASCON '06 : Proceedings of the 2006 conference of*

- the Center for Advanced Studies on Collaborative research*, pages 1–13. IBM, 2006.
- [Hut95] Edwin Hutchins. *Distributed Cognition*. MIT Press, Cambridge, MA, 1995.
- [JW01] Daniel Jackson and Allison Waingold. Lightweight extraction of object models from bytecode. *IEEE Transactions on Software Engineering*, 27(2) :156–169, 2001.
- [KKB+98] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lispon, and J. Carriere. The architecture tradeoff analysis method. In *ICECCS '98 : Proceedings of the Fourth IEEE International Conference on Engineering of Complex Computer Systems, 1998*, pages 68–78. IEEE Computer Society, 1998.
- [KR98] C. Kostelnick and D. D. Roberts. *Designing Visual Language : Strategies for Professional Communicators*. Allyn and Bacon, Boston, 1998.
- [LK98] Anthony Lauder and Stuart Kent. Precise visual specification of design patterns. In *ECOOP '98 : Proceedings of the 12th European Conference on Object-Oriented Programming*, pages 114–134. Springer-Verlag, 1998.
- [Mal08] Jonathan I. Maletic. Ieee international conference on program comprehension (icpc) web site. <http://www.program-comprehension.org>, Avril 2008.
- [MF93] P. Moore and C. Flitz. Gestalt theory and instructional design. *Journal of Technical Writing and Communication*, 23(2) :137–157, 1993.
- [MHG02] David Mapelsden, John Hosking, and John Grundy. Design pattern modelling and instantiation using dpml. In *CRPIT '02 : Proceedings of the Fortieth International Conference on Tools Pacific*, pages 3–11. Australian Computer Society, Inc., 2002.

- [PCA02] Helen C. Purchase, David A. Carrington, and Jo-Anne Alder. Empirical evaluation of aesthetics-based graph layout. *Empirical Software Engineering*, 7(3) :233–255, 2002.
- [Ray98] K. Rayner. Eye movements in reading and information processing : 20 years of research. *Psychological Bulletin*, 124(3) :372–422, November 1998.
- [SK98] R. Schauer and R. Keller. Pattern visualization for software comprehension. In *IWPC '98 : Proceedings of the 6th International Workshop on Program Comprehension*, pages 4–12. IEEE Computer Society, 1998.
- [ST02] Alan Shalloway and James R. Trott. *Design patterns explained : a new perspective on object-oriented design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [SW05] Dabo Sun and Kenny Wong. On evaluating the layout of uml class diagrams for program comprehension. In *IWPC '05 : Proceedings of the 13th International Workshop on Program Comprehension*, pages 317–326. IEEE Computer Society, 2005.
- [TT07] Tim Trese and Scott Tilley. Documenting software systems with views V : towards visual documentation of design patterns as an aid to program understanding. In *SIGDOC '07 : Proceedings of the 25th annual ACM international conference on Design of communication*, pages 103–112. ACM, 2007.
- [Vli98] John Vlissides. Notation, notation, notation. C++ Report, April 1998.
- [War05] Colin Ware. Visual queries : The foundation of visual thinking. In *Knowledge and Information Visualization*, pages 27–35, 2005.
- [WRH⁺00] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering : an introduction*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

- [YKM07] Shehnaaz Yusuf, Huzefa Kagdi, and Jonathan I. Maletic. Assessing the comprehension of uml class diagrams via eye tracking. In *ICPC '07 : Proceedings of the 15th IEEE International Conference on Program Comprehension*, pages 113–122. IEEE Computer Society, 2007.