

Titre: Donner le pouvoir des patrons de conception d'IHM aux développeurs logiciels via les modes de présentation : une application aux logiciels de contrôle de simulateurs de vol
Title:

Auteur: Jocelyn Richard
Author:

Date: 2011

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Richard, J. (2011). Donner le pouvoir des patrons de conception d'IHM aux développeurs logiciels via les modes de présentation : une application aux logiciels de contrôle de simulateurs de vol [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/579/>
Citation:

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/579/>
PolyPublie URL:

Directeurs de recherche: Jean-Marc Robert
Advisors:

Programme: Maîtrise recherche en génie industriel
Program:

UNIVERSITÉ DE MONTRÉAL

DONNER LE POUVOIR DES PATRONS DE CONCEPTION D'IHM AUX
DÉVELOPPEURS LOGICIELS VIA LES MODES DE PRÉSENTATION : UNE
APPLICATION AUX LOGICIELS DE CONTRÔLE DE SIMULATEURS DE VOL

JOCELYN RICHARD

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INDUSTRIEL)

AVRIL 2011

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

DONNER LE POUVOIR DES PATRONS DE CONCEPTION D’IHM AUX DÉVELOPPEURS
LOGICIELS VIA LES MODES DE PRÉSENTATION : UNE APPLICATION AUX
LOGICIELS DE CONTRÔLE DE SIMULATEURS DE VOL

Présenté par : RICHARD Jocelyn

en vue de l’obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d’examen constitué de :

M. FRAYRET Jean-Marc, Ph. D., président

M. ROBERT Jean-Marc, Doctorat, membre et directeur de recherche

M. SEFFAH Ahmed, Ph. D., membre

*Un homme muni de papier, d'un crayon et d'une gomme, sujet à une discipline stricte, est
effectivement une machine universelle.*

Alan Turing

REMERCIEMENTS

Je souhaiterais remercier la compagnie CAE Inc. pour les ressources humaines, matérielles et financières qui m'ont permis de mener ce projet à bien. Merci à M. Sylvain Caron pour sa supervision bienveillante, aux membres du comité des architectes pour leurs retours et leurs conseils avisés ainsi qu'à tous les membres du personnel de CAE qui ont pris de leur temps pour partager leur expertise ou participer aux expérimentations. Merci tout spécialement à M. Sébastien Malo et M. Joël Migneault, spécialistes des facteurs humains, qui m'ont accueilli dans leur équipe et accompagné de leur professionnalisme, de leur sympathie et de leur confiance tous les jours pendant un an.

Je voudrais remercier mon directeur de recherche, M. Jean-Marc Robert, pour son soutien, sa vigilance et son enthousiasme ; ce travail n'existerait pas sans lui.

Je voudrais également remercier mes amis de Polytechnique, qui m'ont aidé à plus d'un titre : M. Philippe Doyon-Poulin, M. Alexandre Galliez et M. Grégory Petit.

Un grand merci enfin à mes parents, mon frère, mon chien Thaïs et toute ma famille, toujours présents pour moi même quand ils sont aux quatre coins du globe.

RÉSUMÉ

Les simulateurs de vols sont une combinaison de matériels et de logiciels sophistiqués destinés à former et maintenir à jour les pilotes d'appareils civils ou militaires dans des conditions sécuritaires et économiques. La conception des interfaces utilisateur de ces simulateurs de vol est une activité complexe et coûteuse dont la difficulté va croissant avec les innovations technologiques : interactions tactiles, moniteurs multiples, simulations multi-joueurs... Plusieurs corps de métiers y prennent part : des experts en ingénierie des facteurs humains mais aussi des développeurs logiciels, non formés en ergonomie.

La maîtrise et l'optimisation du processus de conception passent par la gestion de la connaissance. Les patrons de conception, des artefacts fournissant de manière formelle des solutions génériques à des problèmes de conception récurrents, semblent être particulièrement adéquats pour relever ce défi. C'est toutefois un artefact rédigé par et pour les experts en ingénierie des facteurs humains, de sorte que les développeurs logiciels ont du mal à s'en servir, se privant ainsi d'un outil profitable. La mise à disposition de patrons de conception d'interfaces utilisateur à des développeurs logiciels est un sujet peu étudié à ce jour ; l'étude de stratégies pour y parvenir reste à faire.

Dans ce mémoire, nous effectuons diverses études au sein de la compagnie CAE Inc. pour mettre en place une bibliothèque de patrons de conception et la rendre accessible aux développeurs logiciels comme aux ergonomes.

Nous avons commencé par étudier et comparer les bibliothèques de patrons de conception d'interfaces utilisateur les plus populaires. Nous avons ensuite rédigé une trentaine de patrons de conception couvrant les logiciels de contrôle de simulateurs de vol de toutes sortes, nous inspirant à la fois de la revue de littérature et du travail de terrain au sein de l'entreprise. Ces patrons ont ensuite été diffusés de manière restreinte sur le réseau intranet de l'entreprise, via un système de gestion de contenu en source libre modifié pour subvenir à nos besoins. Ils ont ainsi été lus, utilisés et validés par une cinquantaine de professionnels de CAE dont l'équipe d'ingénierie des facteurs humains.

Nous avons ensuite mis au point de nouvelles manières de présenter ces patrons aux développeurs logiciels : les types d'applications, arbres de décision et aperçus de patrons. Ces

modes de présentation ont été comparés au mode de présentation de référence lors d'une expérimentation dans l'entreprise, avec des participants novices n'ayant jamais utilisé de patrons de conception d'interfaces utilisateur en générale ni la bibliothèque de patrons de CAE en particulier. Nos résultats montrent que les développeurs logiciels trouvent des patrons 1,5 à deux fois plus pertinents avec les nouveaux modes de présentation, tout en étant en moyenne 50% plus satisfaits. Ces gains se font cependant au prix d'une augmentation du temps de recherche des patrons pouvant aller jusqu'à 50%. Même si notre étude expérimentale est d'une ampleur modérée et qu'il faudrait recueillir plus de données pour l'étendre à d'autres cas, nous pouvons recommander l'emploi de modes de présentation alternatifs dans une bibliothèque de patrons de conception d'interfaces utilisateur.

Ces résultats sont synthétisés en recommandations immédiatement applicables dans un environnement professionnel et des actions à prendre en complément au niveau technique ou organisationnel sont suggérées.

ABSTRACT

Flight simulators are a blend of sophisticated hardware and software systems. They are intended to train and keep pilots of civilian or military aircrafts up-to-date in safe and economical conditions. User interface design for flight simulators is a complex and expensive process whose difficulty is increasing along technological innovation: touch interactions, multiple monitors, multiplayer simulations... Several building trades take part in this endeavor: human factors engineering experts as well as software developers, the latter lacking formal usability training.

The control and optimization of the design process go through knowledge management. Design patterns are formal generic solutions to recurring design problems; they seem especially suitable to take on this challenge. They are however being written by and for human factors engineering experts, hence hard to use for software developers that are therefore denied a profitable tool. Making user interface design patterns available to software developers is a sparsely studied topic; no strategies have yet been defined to this end.

In this thesis, we carry out various studies among CAE Inc. in order to set up a design pattern library and make it accessible to both software developers and usability practitioners.

We started by studying and comparing the most popular user interface design pattern libraries. We then wrote thirty design patterns covering flight simulator control software of all kinds, building upon our review of literature and field work at CAE. Those patterns have then been privately shared on the intranet network of the company via an open-source content management system, tweaked to fit our needs. They have thus been read, used and validated by around fifty CAE professionals, including the human factors engineering team.

We subsequently developed new ways of presenting those patterns to software developers: application types, decision trees and patterns previews. Those presentation modes were compared to a reference presentation mode during a test session in the company. Participants were beginners that had never used user interface design patterns and had no prior knowledge of CAE's pattern library. Our results show that software developers find 1.5 to two times more relevant patterns with the new presentation modes, while being in average 50% more satisfied. However, those gains come at the expense of speed since search durations increase up to 50%.

Even though our experimentation has a limited scope, we can recommend the use of alternative presentation modes in user interface design pattern libraries.

Those results are summed up and turned into recommendations for immediate use in a professional environment. Complementary actions for technical and organizational matters are suggested.

TABLE DES MATIÈRES

REMERCIEMENTS	IV
RESUME	V
ABSTRACT	VII
TABLE DES MATIERES	IX
LISTE DES TABLEAUX	XIII
LISTE DES FIGURES	XIV
LISTE DES SIGLES ET ABRÉVIATIONS	XV
LISTE DES ANNEXES	XVI
INTRODUCTION	1
CHAPITRE 1 REVUE DE LITTÉRATURE SUR LES BIBLIOTHÈQUES DE PATRONS ET LEUR USAGE PAR LES DÉVELOPPEURS LOGICIELS	6
1.1 La gestion de la connaissance	6
1.1.1 Lignes directrices	7
1.1.2 Composants	9
1.1.3 Les patrons de conception	10
1.1.4 Comparatif	19
1.2 Les bibliothèques de patrons de conception d’interfaces utilisateur	20
1.2.1 Méthodologie	21
1.2.2 Résultats	22
1.2.3 Analyse	22
1.3 Les difficultés des développeurs logiciels	28
1.3.1 Pas de guidage dans les bibliothèques de patrons	29
1.3.2 Pas de conseil lors de la lecture d’un patron	30

1.3.3	La variabilité des schémas de patron	31
1.3.4	Le manque d'outils informatiques	31
1.3.5	Conclusion	32
1.4	Synthèse de la revue de littérature	32
CHAPITRE 2 PROBLÉMATIQUE, OBJECTIFS ET DÉVELOPPEMENT D'UNE BIBLIOTHÈQUE DE PATRONS POUR INTERFACES UTILISATEUR		34
2.1	Question et activités de recherche	34
2.2	Hypothèses	35
2.2.1	Justification de l'originalité	35
2.2.2	Réfutabilité	35
2.3	Travaux préparatoires	35
2.3.1	L'automatisation mise de côté	36
2.3.2	Des relations entre patrons <i>a minima</i>	38
2.4	Rédaction des patrons	38
2.4.1	Recensement des besoins et des bonnes pratiques	38
2.4.2	Rédaction d'un matériel adapté aux développeurs logiciels	40
2.4.3	Développement du système de gestion de contenu	41
2.4.4	Itération et finalisation	41
2.5	Développement des nouveaux modes de présentation	43
2.5.1	Types d'applications	43
2.5.2	Arbres de décision	48
2.5.3	Aperçus de patrons	50
2.5.4	Vérification et validation	52
CHAPITRE 3 ÉVALUATION EXPÉRIMENTALE DES NOUVEAUX MODES DE PRÉSENTATION		55

3.1	Méthodologie	56
3.1.1	Participants	56
3.1.2	Tâches	56
3.1.3	Écrans choisis par les participants.....	57
3.1.4	Patrons de conception	57
3.1.5	Système informatique utilisé	58
3.1.6	Données recueillies	58
3.1.7	Procédure	59
3.2	Résultats	59
3.2.1	Taux de succès dans la recherche de patrons	59
3.2.2	Temps de recherche.....	59
3.2.3	Satisfaction des utilisateurs	61
3.2.4	Pertinence des patrons de conception trouvés	62
3.2.5	Corrélation des résultats expérimentaux avec les éléments fournis par le questionnaire biographique	63
CHAPITRE 4	DISCUSSION DES RÉSULTATS.....	65
4.1	Interprétation des résultats	65
4.1.1	Temps requis pour trouver un patron.....	65
4.1.2	Arbres de décision et types d'application	66
4.1.3	Aperçus de patrons.....	67
4.1.4	Rédaction adaptée aux développeurs	68
4.2	Recommandations	68
4.2.1	Commencer chaque patron par son aperçu.....	68
4.2.2	Supporter la bibliothèque dès que possible avec des types d'applications et des arbres de décision.....	69

4.2.3	Garder un index alphabétique disponible	70
4.2.4	Écrire pour un public néophyte	70
4.2.5	Désigner un rédacteur en chef	72
4.2.6	Mettre de côté des ressources pour « l'utilisabilité de l'utilisabilité »	74
4.3	Actions à prendre.....	74
4.3.1	Continuer à alimenter la bibliothèque en patrons.....	74
4.3.2	Évangéliser et rassurer	75
4.3.3	Corriger et enrichir les méthodes mises en place	77
4.3.4	Transformer la bibliothèque en langage	80
CONCLUSION		85
Pistes de recherche & travaux futurs		86
BIBLIOGRAPHIE		88
ANNEXES		96

LISTE DES TABLEAUX

Tableau 1-1: Exemples de lignes directrices, triés par niveau d'abstraction décroissant.....	7
Tableau 1-2 : Caractéristiques des langages de patrons	14
Tableau 1-3: Comparaison entre lignes directrices, composants et patrons de conception	20
Tableau 1-4 : Le processus des 7C.....	27
Tableau 1-5 : Flux des tâches pour la création de patrons	28
Tableau 2-1: Échelle de spécificité des patrons de conception CAE	42

LISTE DES FIGURES

Figure 0-1: Gamme de solutions CAE.....	1
Figure 1-1: Ligne directrice de type <i>consigne d'utilisation</i> pour Windows 7 (extraits)	8
Figure 1-2: Composants du site CNN.com.....	10
Figure 1-3: Patron <i>Transition autoguérissante</i>	13
Figure 1-4: Mode de présentation des patrons du chapitre <i>Making it look good</i>	25
Figure 2-1: Postures pour sites Web de magasinage en ligne	44
Figure 2-2: Standard Screen Patterns (extraits)	45
Figure 2-3: Type d'applications <i>Modèle Immersif</i>	47
Figure 2-4: Arbre de décision <i>Limiter le taux d'erreur</i> (assistant, extraits).....	49
Figure 2-5: Arbre de décision <i>Limiter le taux d'erreur</i> (diagramme)	50
Figure 2-6: Aperçu du patron <i>Transition plein-écran</i>	52
Figure 3-1: Temps requis (en minutes) pour trouver un patron	60
Figure 3-2: Satisfaction moyenne des utilisateurs	62
Figure 3-3: Pertinence moyenne des patrons trouvés	63
Figure 4-1: Rapport de la satisfaction des utilisateurs sur le temps de recherche des patrons.....	66
Figure 4-2: Composantes d'une bibliothèque et d'un langage de patrons.....	80
Figure 4-3: Interdépendance des activités de conception centrée sur l'opérateur humain	83

LISTE DES SIGLES ET ABRÉVIATIONS

HTML	HyperText Markup Language
IHM	Interface Homme-Machine
JSON	JavaScript Object Notation
PHP	Hypertext Preprocessor
PCB	Pattern- and Component-based Process
PLML	Pattern Language Markup Language
POD	Pattern Oriented Design
PSA	Pattern Supported Approach
WAMP	Windows, Apache, MySQL, PHP
XML	eXtensible Markup Language

LISTE DES ANNEXES

ANNEXE A – REVUE DE LITTÉRATURE	96
A.1 Présentation des bibliothèques revues	97
A.2 Public visé, thèmes couverts et relations entre patrons.....	98
A.3 Modes de présentation & lisibilité machine.....	99
A.4 Exemples de patrons.....	100
A.5 Visualisation des modes de présentation	102
ANNEXE B – SYSTÈME DE GESTION DE CONTENU.....	108
B.1 Solutions comparées	108
B.2 Configuration de la solution choisie	109
ANNEXE C – BIBLIOTHÈQUE DE PATRONS CAE	110
C.1 Liste des patrons CAE.....	110
C.2 Types d’application CAE	111
C.3 Arbres de décision CAE	111
C.4 Exemple de patron CAE.....	112
C.5 Visualisation des modes de presentation CAE.....	115
ANNEXE D – ÉVALUATION EXPÉRIMENTALE	116
D.1 Certificat d’acceptation d’un projet de recherche par le comité d’éthique	116
D.2 Questionnaire biographique	117
D.3 Questionnaire post-expérimentation.....	117
D.4 Participants	118
D.5 Patrons trouvés.....	119
D.6 Ordre du choix des modes de présentation par les participants	120
D.7 Pertinence des patrons trouvés.....	121

D.8 Taux de satisfaction & mode de présentation favori des participants	122
D.9 Temps de recherche pour trouver un patron approprié	123
D.10 Variation de la satisfaction des utilisateurs relativement à la variation du temps de recherche d'un patron approprié	123
D.11 Normalisation de la pertinence des patrons trouvés et corrélation avec les catégories de participant.....	124
D.12 Normalisation du taux de satisfaction des participants et corrélation avec les catégories de participant.....	126
D.13 Normalisation du temps de recherche des patrons et corrélation avec les catégories de participant.....	128

INTRODUCTION

CAE est un chef de file mondial dans le domaine des technologies de simulation et de modélisation et des solutions intégrées de formation destinées à l'aviation civile et aux forces de défense du monde entier. Elle génère des revenus annuels de plus de 1,5 milliards de dollars canadiens et compte plus de 7 500 employés dans plus de 100 sites et centres de formation répartis dans plus de 20 pays. Plus de 80 000 membres d'équipage s'entraînent chaque année dans son réseau mondial de 30 centres de formation civils et militaires. (CAE, 2010a)

Les outils de simulation permettent de former du personnel dans les domaines où la sécurité est cruciale et la marge d'erreur est minime, la complexité et les coûts de formation sur les appareils réels sont élevés (CAE, 2010b). Ils peuvent prendre la forme de logiciels exécutés sur des ordinateurs grand public, de systèmes logiciels et matériels statiques pour quelques personnes ou habitacles dynamiques complets impliquant plusieurs dizaines d'opérateurs (figure 0-1).



Figure 0-1: Gamme de solutions CAE.

La formation sur simulateur présente de nombreux avantages par rapport à la formation sur appareil réel. Elle est notamment plus sécuritaire¹, plus économique, jusqu'à 1/10^{ème} du coût de la formation sur un appareil réel, et plus pédagogique via le contrôle des variables et des conditions d'exercice.

Le développement de ces outils de simulations est un chantier transversal faisant intervenir de nombreux corps de métiers. Même la seule partie logicielle (les logiciels de contrôle de simulateurs de vol), objet de ce mémoire, implique une multitude de profils : depuis les architectes logiciels à la barre des grands choix technologiques de l'entreprise jusqu'aux propriétaires de produits gérant le personnel, le budget et les délais, en passant par les développeurs logiciels programmant la logique des simulateurs. Dans une entreprise comme CAE travaillant sur des gros projets, pour moitié dans la défense, la programmation logicielle présente d'ailleurs des caractéristiques singulières :

- La complexité des architectures logicielles. Les systèmes requis pour une simulation « complète et hautement immersive » nécessitent une immense puissance de calcul, des traitements concurrents et distribués, avec possibilité de délais entre serveurs, une multitude de technologies et de langages de programmation...
- La variabilité des projets. Le coût et le cadre d'usage particulier des outils de simulation les destinent naturellement à des fabrications plus unitaires que de grande série. Comme en plus les produits sont souvent personnalisés en fonction du client, la réutilisation de ressources paraît a priori difficile.

Cette complexité et cette variété se répercutent sur l'*interface utilisateur* des outils de simulation, qui doivent accommoder des saisies par clavier/souris ou par écran tactile (au stylet ou à la main), des affichages oscillant entre un moniteur de 12 pouces à 60 pouces et six moniteurs de 19 pouces, des contextes d'usage variant du confort d'une salle de classe à la rigueur d'un entraînement de Marines...

Pour ajouter à ce foisonnement de contraintes, les clients sont de plus en plus exigeants envers leurs produits ou services interactifs. Norman (2002) va jusqu'à proclamer que l'utilisabilité sera

¹ Marc Parent, PDG de CAE Inc., évoquait par exemple le besoin de s'entraîner avant de pratiquer « un ravitaillement en vol de nuit au-dessus du Kosovo » lors d'un colloque interne en 2010.

le prochain terrain d'affrontement entre les entreprises et les succès récents de Apple, Audi ou Virgin Atlantic, attribuables en partie à la qualité de leurs expériences utilisateurs, tendent à confirmer cette affirmation.

Au final, les coûts de conception des outils de simulation sont colossaux et vont croissant. La nécessité de maîtriser ces coûts s'accompagne de la nécessité d'augmenter la qualité des Interfaces Homme-Machine (IHM). Deux corps de métiers interviennent à ce niveau :

- L'équipe d'ingénierie des facteurs humains de CAE (que nous appellerons également *ergonomes*), conçoit l'IHM et la livre sous forme de spécifications, maquettes, prototypes, ou expertise... Elle définit l'essentiel, mais pas forcément la totalité, faute de temps ou de spécifications initiales exhaustives, de l'interface des logiciels.
- Les développeurs logiciels transforment cette conception en logiciel effectif. L'implémentation des maquettes, bien que suivie par les ergonomes de CAE, reste au final à la discrétion des développeurs. *Développeurs logiciel* est un terme parapluie englobant une multitude de postes techniques, tels que technicien, ingénieur, spécialiste ou architecte logiciel, ingénieur système, gestionnaire d'équipe ou gestionnaire de projet. Tous ont un diplôme d'études universitaires en Génie Logiciel ou en Génie Informatique et effectuent des tâches de développement logiciel dans l'entreprise, avec divers niveaux d'expérience : programmation, installation, configuration, tests, correction de bogues, maintenance...

Comme le nombre d'ergonomes chez CAE est inférieur au nombre de développeurs dans un rapport de un à 20, leur disponibilité pour les équipes de développement peut être insuffisante. Nombre de développeurs préfèrent ne pas attendre pour avancer sur leur produit ; d'autres ne sont tout simplement pas sensibilisés ou intéressés par les aspects *utilisabilité* ou *expérience utilisateur* du développement logiciel. La conséquence est la même dans les deux cas : une partie des IHM de l'entreprise est réalisée par des gens qui n'ont pas été formés en ergonomie. Il en résulte une diminution de la qualité et de la productivité.

CAE est conscient de cet état de fait et désireux d'améliorer la qualité de ses IHM tout en réduisant les coûts de développement. L'équipe d'ingénierie des facteurs humains a identifié les patrons de conception comme un levier potentiel à même de multiplier l'action des ergonomes de

l'entreprise auprès des développeurs logiciels. C'est dans ce cadre qu'ils ont accueilli l'auteur de ce mémoire pour un stage d'un an².

Les patrons de conception stockent et partagent la connaissance sous la forme de texte agrémenté d'images. Granlund, Lafrenière & Carr (2001) les définissent comme des conseils puissants et génériques, transmettant la connaissance sur les bonnes pratiques de conception dans un format cohérent, facile à lire et à comprendre, utilisables par les ergonomes comme par les développeurs logiciels. Les patrons sont pratiques pour les développeurs car ils résolvent des problèmes de haut niveau sans nécessiter une connaissance approfondie de principes ou de lignes directrices complexes (Stapleton, 2007). Ils ont pour effet d'après Lammi (2007) d'améliorer l'homogénéité et la qualité des interfaces utilisateur, de rendre la conception et l'implémentation plus efficaces, fournissant un avantage compétitif à l'entreprise. Ils sont d'ordinaire regroupés en bibliothèques. Une telle bibliothèque, portant sur les logiciels de contrôle de simulateur de vol, semble être le candidat tout indiqué pour traiter les problèmes de CAE.

Dans les faits cependant, les patrons de conception sont plutôt utilisés par les ergonomes. Leur usage reste faible chez les développeurs logiciels, en dépit de la grande quantité de bibliothèques de patrons disponibles (Mahemoff & Johnston, 2001), pour plusieurs raisons : un manque d'organisation des patrons, leur complexité et un manque d'outils pour supporter les développeurs logiciels et alléger leur charge de travail.

Ces difficultés risquent de compromettre les efforts entrepris par CAE : si les développeurs logiciels n'arrivent pas à se servir des patrons de conception d'interfaces utilisateur, alors l'entreprise n'obtiendra pas les gains promis et ne sera pas en mesure de résoudre son problème initial de productivité et de qualité. Le travail rapporté dans ce mémoire porte donc sur la mise en place d'une bibliothèque de patrons de conception pour logiciels de contrôle de simulateurs de vol et sur les mesures prises pour la rendre accessible autant aux développeurs logiciels qu'aux ergonomes. Ces mesures concernent essentiellement les *modes de présentation*, c'est-à-dire les points d'entrée guidant les utilisateurs vers les patrons de la bibliothèque. Ce sont des index recourant éventuellement à différentes modalités de dévoilement progressif, d'assistance ou de découpage de la bibliothèque en sous-ensembles thématiques. Dans notre cas, il s'agit de mettre

² D'avril 2010 à avril 2011.

en place de nouveaux modes de présentation adaptés aux développeurs logiciels. Enfin, les patrons seront rédigés selon un style adapté à un public de développeurs logiciels.

Le reste du mémoire est organisé comme suit :

Le chapitre 1 contient une revue de littérature des bibliothèques de patrons utilisés en conception des IHM, détaillant notamment leurs modes de présentation, et un florilège des difficultés qu'ont les développeurs logiciel à les utiliser.

Le chapitre 2 énonce la question de recherche, ainsi que les hypothèses et les objectifs s'y rapportant. Il expose également le développement de la bibliothèque de patrons de CAE : vision et choix technologiques et fonctionnels, rédaction des patrons, adaptation d'un système de gestion de contenu pour héberger la bibliothèque et élaboration de modes de présentation innovants pour les patrons.

Le chapitre 3 présente la méthodologie et les résultats de l'évaluation expérimentale des innovations proposées.

Le chapitre 4 discute les résultats précédemment obtenus et en tire des recommandations immédiatement applicables ainsi que la marche à suivre pour continuer à développer la bibliothèque de patrons de CAE.

La conclusion résume le contexte du travail, synthétise les contributions apportées et propose des pistes de recherche complémentaires.

CHAPITRE 1 REVUE DE LITTÉRATURE SUR LES BIBLIOTHÈQUES DE PATRONS ET LEUR USAGE PAR LES DÉVELOPPEURS LOGICIELS

Ce chapitre présente la revue de littérature que nous avons menée pour connaître l'état actuel des connaissances à propos des différents artefacts de gestion de la connaissance et des bibliothèques de patrons de conception d'interfaces utilisateur existantes, en portant une attention particulière à la manière dont leurs patrons sont rédigés et présentés. Des données sur les difficultés des développeurs logiciels lors de l'utilisation des patrons de conception d'interfaces utilisateur sont recueillies en vue de comprendre le problème qu'ils éprouvent.

Ces éléments vont nous permettre de partir d'un état de l'art connu pour progresser en exploitant les connaissances existantes.

1.1 La gestion de la connaissance

La gestion de la connaissance consiste à acquérir, créer, organiser, distribuer et utiliser des ressources intellectuelles (Davenport, 1995; Grover & Davenport, 2001). C'est un facteur fondamental de la compétitivité, d'autant plus important dans les entreprises comme CAE dont l'essentiel de l'activité et du patrimoine est immatériel. La gestion de la connaissance permet de remédier en partie à l'escalade des coûts de développement via la documentation et la réutilisation de code informatique, capitalisant sur les travaux passés plutôt que les réinventer à chaque fois, de façon à permettre de se concentrer sur les aspects novateurs (Spool, 2009b).

CAE dispose d'un portfolio de produits variés, sophistiqués et très spécifiques, aux interfaces multiformes. L'entreprise est donc un bon candidat à la gestion de la connaissance.

Nous nous intéresserons au domaine des interfaces humain-ordinateur en particulier. En effet, l'interface utilisateur représente déjà 60% des coûts de développement logiciels (Myers & Rosson, 1991), et la tendance est à la hausse avec l'accroissement des exigences des consommateurs ou l'avènement de produits hautement interactifs comme les téléphones intelligents, mais sa conception demeure très empirique comparativement à d'autres disciplines. La gestion de la connaissance pourrait contribuer à transformer cet artisanat en science. Nous nous intéresserons ici aux artefacts de gestion de la connaissance les plus répandus, c'est-à-dire

les lignes directrices, les composants et les patrons de conception dont nous exposerons les différences d'objectifs, de nature et de cas d'usage, avant de les comparer dans un tableau récapitulatif.

1.1.1 Lignes directrices

Les lignes directrices sont l'artefact de gestion de la connaissance le plus répandu dans le domaine des IHM. Elles prennent la forme de texte libre généralement illustré par des captures d'écran. Elles garantissent l'utilisabilité et surtout la cohérence fonctionnelle et esthétique d'une plate-forme, d'une organisation ou d'un système interactif par rapport à lui-même ou au sein d'une gamme de produits. Les lignes directrices peuvent couvrir des sujets allant du très abstrait au très concret (tableau 1-1).

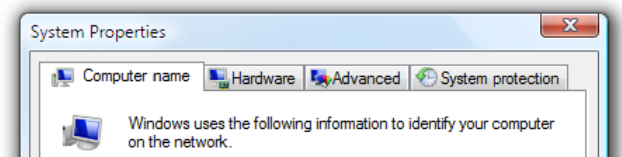
Tableau 1-1 : Exemples de lignes directrices, triés par niveau d'abstraction décroissant

Nom	Niveau d'abstraction	Extrait
Ask Tog	Principe général	« Homogénéité »
Usability.gov - Accessibility	Objectif à atteindre	« Les sites Web devraient être conçus de sorte que tout le monde puisse s'en servir. »
UI Design and Interaction Guide for Windows Phone 7	Vision à long terme	« L'interface reflète le langage visuel des aéroports et des métros dans son apparence et sa typographie. »
Jakob Nielsen	Heuristique	« Reconnaissance plutôt que rappel. »
Apple Human Interface Guidelines	Consignes d'utilisation	« Utilisez des onglets quand vous voulez présenter un petit nombre de vues du contenu à un seul endroit dans une fenêtre. »
webOS User Interface Guidelines	Conventions et recommandations	« La police de caractères du contenu de votre application ne devrait jamais être plus petite que 15 pixels ; les libellés ne devraient jamais être plus petits que 14 pixels. »

Les plus abstraites peuvent donner des principes d'utilisabilité généraux, des objectifs à atteindre ou la vision du produit à long terme. D'autres lignes directrices, plus appliquées, guident directement les décisions de conception par des consignes concernant les éléments « physiques » affichés à l'écran tels que des boîtes de dialogue, menus, texte, cases à cocher ou autres boutons radio (figure 1-1). Enfin les lignes directrices les plus pratiques spécifient les conventions et recommandations idiosyncratiques à une entreprise, reflétant ses besoins et sa terminologie ; elles forment un corpus généralement nommé guide de style. La conformité d'un produit au guide de style de l'entreprise est d'ordinaire obligatoire.

- **Don't put icons on tabs.** Icons usually add unnecessary visual clutter, consume screen space, and often don't improve user comprehension. Only add icons that aid in comprehension, such as standard symbols.

Incorrect:



In this example, the icons add visual clutter and do little to improve user comprehension.

Exception: You can use clearly recognizable icons if there might be insufficient space to display meaningful labels:

Correct:

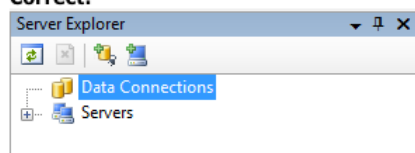


Figure 1-1: Ligne directrice de type *consigne d'utilisation* pour Windows 7 (extraits)

Les lignes directrices sont critiquées à plusieurs niveaux.

Les plus abstraites sont considérées difficiles à interpréter, et ne peuvent pas être utilisées efficacement par des gens qui ne sont pas des experts en facteurs humains (Mahemoff & Johnston, 1998). En visant la cohérence des interfaces par la documentation des complications d'une plate-forme en particulier (comme Windows XP ou le langage Java), les lignes directrices plus appliquées se concentrent sur des problématiques de placement de contrôles à l'écran, négligeant le savoir plus large nécessaire à la conception d'une bonne interface. En fait, des facteurs primordiaux à prendre en compte lors de la conception, comme l'utilisateur, le contexte,

le cas d'usage ou la tâche, sont absents des lignes directrices (Granlund et al., 2001). Cette même notion d'association à une technologie en particulier en fait de mauvais vecteurs de réutilisation : les recommandations sont spécifiques à une technologie et ne sont pas transférables à une autre. La plupart des lignes directrices ne parviennent donc pas à transmettre la connaissance accumulée aux développeurs (Javahery, 2007). Enfin d'un point de vue pratique, les lignes directrices fournissent des informations générales et non structurées, demandant un effort excessif pour trouver la section pertinente (Mahemoff & Johnston, 1998). Faire connaître toutes ces informations aux développeurs est une bataille perdue d'avance (Stapleton, 2011).

Au final, les lignes directrices n'ont pas réalisé leur potentiel et ont un impact minimal sur la conception des interfaces utilisateur (Javahery, 2007). Elles deviennent souvent un document statique lu uniquement par des ergonomes pour évaluer la conformité de leur interface à des standards d'utilisabilité (Lammi, 2007), une référence générale sur les pratiques de conception plutôt qu'un outil effectif de support à la conception ; c'est une documentation nécessaire mais pas suffisante.

1.1.2 Composants

Les *composants*, parfois appelés *blocs* ou *modules*, sont des morceaux de code informatique définissant les éléments individuels d'une interface utilisateur. Curtis (2008) les décrit comme des structures autonomes, réutilisables, spécifiques à un système et à but unique sur un écran donné.

Les composants sont dits *autonomes* car ce sont les unités les plus granulaires que documente l'organisation, ils n'ont pas besoin d'éléments plus petits pour être définis. Ils sont également autonomes dans le sens où ils sont immédiatement prêts à l'emploi, chaque composant ayant un contexte d'application bien compris et documenté (Spool, 2009a), ne nécessitant pas de réflexion pour être utilisé.

On les qualifie de *réutilisables* car ils prennent la forme de code informatique très formalisé et modulaire, pouvant être tiré au besoin depuis un *dépôt de code*. Si les lignes directrices et les patrons de conception sont la moitié abstraite de la solution, alors les composants en sont la moitié concrète. Ils traduisent des artefacts génériques en code spécifique, on dit qu'ils les *instancient*. Chaque composant est effectivement la *conception de référence*, un modèle destiné à

être copié, d'une organisation pour implémenter une ligne de conduite ou un patron de conception.

Les composants sont *spécifiques* à un système car ils contiennent des éléments de l'image de marque (polices, couleurs, apparence...), des comportements, des formats, libellés ou autres, valides uniquement pour ce système, exactement comme les boutons radio Mac OS sont différents de ceux de Windows 7 et ne pourraient pas y être utilisés.

Enfin leur but est *unique* sur un écran. Les composants ont chacun une application précise et singulière dans un écran en fonction de leur position, de leur importance hiérarchique ou de leur rôle (Curtis, 2009). Par exemple, on observe sur la figure 1-2, adaptée de Curtis (2009), différents types d'onglets pour le site CNN.com, utilisés pour des catégories (par exemple « Asia »), des types de contenus (par exemple « Video ») ou des critères (par exemple « Most popular ») mutuellement exclusifs. Chaque cas respecte les fondamentaux du gabarit, comme le fait de distinguer l'onglet actif des autres onglets. En revanche l'utilisation, la position sur la page et le style sont différents pour les trois composants.

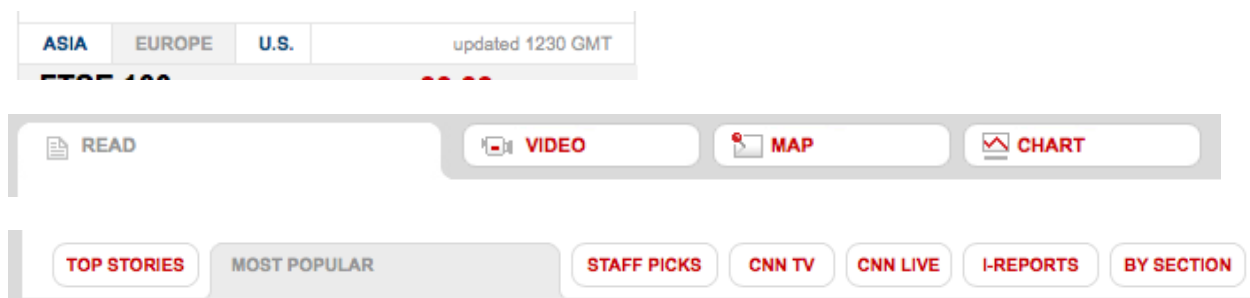


Figure 1-2: Composants du site CNN.com

1.1.3 Les patrons de conception

« Chaque patron décrit un problème qui survient encore et encore dans notre environnement, et décrit l'essence de la solution à ce problème, de manière à pouvoir utiliser cette solution un million de fois, sans jamais faire deux fois la même chose »
Traduction libre du texte d'Alexander, Ishikawa & Silverstein (1977), page x.

Cette section présente en détail les patrons de conception, explique comment les organiser en bibliothèque de patrons ou en langage de patrons via un système de valeur et un principe

structurant. Les avantages des patrons de conception par rapport à d'autres artefacts de gestion de la connaissance seront également décrits.

1.1.3.1 Contexte historique

Mathématicien de formation, Alexander voit l'architecture au milieu des années 60 comme une pratique analytique décomposant chaque problème en sous-problèmes, eux-mêmes exprimables par un ensemble de forces en conflits. En résolvant individuellement ces conflits puis en combinant leurs solutions, l'architecte obtient une solution au problème global. Alexander va jusqu'à évoquer un support informatique lorsqu'il expose cette approche dans Notes on the Synthesis of Form (Alexander, 1964). Mais confronté aux pratiques de l'architecture moderne comme le Modernisme ou le Style International, Alexander juge que de tels processus de conception rigides amènent des solutions peu pratiques, impersonnelles et appauvrissantes pour l'environnement ; il commence à douter de son idée originale.

Il change complètement d'avis après une décennie de recherche. Alexander et son équipe étudient longuement la manière dont les populations aménagent empiriquement leur habitat, particulièrement les cultures anciennes ayant affiné leurs pratiques au fil des générations, et observent qu'elles tendent à répondre aux mêmes problèmes par les mêmes solutions.

Sa nouvelle hypothèse avance que tous les bâtiments agréables et fonctionnels pour leurs occupants et enrichissants pour leur communauté partagent des caractéristiques invariantes et récurrentes dans toutes les cultures, à toutes les époques et dans une variété de contextes. Il capture ces invariants sous une forme abstraite, granulaire, prescriptive et simple à comprendre (Tidwell, 1999) qu'il nomme patrons de conception, et en rassemble 253 dans son essai A Pattern Language (Alexander et al., 1977) :

« Chaque patron est énoncé de telle manière qu'il donne les champs de relation essentiels nécessaires pour résoudre le problème, mais d'une manière très générale et abstraite—de sorte qu'on puisse résoudre le problème par soi-même, à sa manière, en l'adaptant à ses préférences, et aux conditions locales de l'endroit où on le fait » Traduction libre du texte de la page xiii.

Un second tome, The Timeless Way of Building (Alexander, 1979), explique comment combiner les patrons précédents entre eux pour générer des solutions architecturales complètes, c'est la notion de langage de patrons. Ces deux ouvrages fondateurs sont moins destinés aux architectes

qu'aux hommes du commun, à qui ils donnent un « mode d'emploi » pour réutiliser la connaissance et les bonnes pratiques architecturales.

Ce travail marque un jalon dans la pratique de l'architecture : il reçoit un accueil polémique, mais retentissant de la part de la profession³. Les ouvrages d'Alexander capturent l'imagination du public, influençant plusieurs disciplines scientifiques (Dearden & Finlay, 2006). Moins de 10 ans après la publication de The Timeless Way of Building, Cunningham & Beck (1987) transposent l'idée des patrons au domaine du développement logiciel en l'appliquant au langage de programmation Smalltalk. D'autres efforts sont menés en pédagogie ou en littérature mais c'est en programmation orientée-objet que les événements s'accélèrent avec la publication de Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson & Vlissides, 1994). Cet ouvrage, très bien reçu par les praticiens et fréquemment utilisé dans le monde professionnel, est maintenant enseigné en génie informatique et en génie logiciel. Jenifer Tidwell publie peu après Common ground (Tidwell, 1999), la première bibliothèque de patrons d'interfaces utilisateur de grande ampleur. Là encore c'est un succès, le livre qui en est tiré (Tidwell, 2005) est une référence dans le domaine. Welie (2000) publie en suivant une bibliothèque de patrons pour sites Internet, également très populaire en dépit d'une interprétation plus lâche du concept de patrons. La consécration arrive avec la rédaction et la mise à disposition publique de la bibliothèque de patrons de Yahoo (Leacock, Malone, Wheeler, & Scott, 2006), dont un exemple est montré figure 1-3. Depuis, les bibliothèques de patrons de conception d'interfaces utilisateur se multiplient et s'attaquent à des sujets de niche : les interfaces sociales (Crumlish & Malone, 2009), la recherche (Morville & Callender, 2010), les téléphones intelligents Android (UNITiD, 2010)...

³ Se référer à la critique passionnante écrite par Saunders (2002) dans le journal de Harvard.

Self Healing Transition

Best Practice · Last modified July 15, 2009

When an object is being removed from a list or grid, make a hole for the removed object and then animate a transition as the object below or next to the hole moves to close up the open space.

This transition communicates (and reinforces visually) where the object was removed from and that the list or grid no longer contains the object that was removed.



Buzz up! Bookmark this on Delicious

What Problem Does This Solve?

The designer wants to show that an object has been removed from a list of objects.

When to Use This Pattern

- The object being removed is in a list or grid.
- It is important to know that the removal happened and where the object was removed from the list or grid.
- At the completion of a drop operation where the dropped object was moved from one place to another.
- In conjunction with the object appearing somewhere else (usually with a fade-in).

What's the Solution?

- Make a hole for the removed object.
- When the interaction completes the object below slides up to close up the hole.
- See the [Slide Pattern](#).

Why Use This Pattern?

The self healing transition communicates:

- Where the object was removed from.
- That the list no longer contains the object that was removed.

For more information, see the [rationale](#) for Transition patterns.

Accessibility

See the [general discussion about accessibility](#) for Transition patterns.

Figure 1-3: Patron *Transition autoguérissante*

1.1.3.2 Les langages de patrons

Les patrons de Alexander ne vivent pas en isolation, ils font partie d'un langage. Un langage de patron est un réseau d'interdépendance, formé quand un ensemble de patrons est arrangé en une structure cohésive, les patrons de haut niveau fournissant un contexte résolu par des patrons plus détaillés. Par exemple, le patron *Arrêt de bus* d'Alexander suggère un placement en fonction d'autres patrons comme *Une place pour attendre* ou *Passage principal*.

La solution est atteinte quand le patron choisi et les sous-patrons qui le composent ont tous été appliqués. Le processus est récursif : chaque patron après le premier peut à son tour donner du contexte et des références vers des patrons suivants (Mahemoff & Johnston, 1998). Les règles régissant les connexions entre patrons, ou nœuds, sont aussi importantes que les patrons eux-mêmes.

Comme les praticiens en interfaces humain-ordinateur ne se sont pas arrêtés sur une définition unique des patrons de conception ou des langages de patrons (Vlissides, 1997) et interprètent librement les concepts de Alexander, il n'existe pas de définition de langage de patron ayant valeur de standard. Fincher & Utting (2002) ont proposé neuf qualités nécessaires à l'élaboration d'un langage (tableau 1-2). Nous nous en servons comme référence.

Tableau 1-2 : Caractéristiques des langages de patrons

Requis fonctionnels	Requis non fonctionnels
Capture de la pratique	Remarquable
Abstraction	Compréhension
Système de valeurs	-
Principe structurant	Génératif
Formalisme de présentation	Communicable

- La *capture de la pratique* correspond au fait que les patrons ne capturent pas l'innovation ou les vœux pieux de l'organisation mais se basent fermement sur des solutions éprouvées.
- L'*abstraction* détache les bonnes pratiques de leur exécution technique ou esthétique pour n'en garder que les aspects universels. C'est ce qui assure la possibilité de réutiliser le patron.
- Le *système de valeurs* est la philosophie sous-jacente qui assure la cohésion du langage. Comme les patrons sont l'existant observé, interprété et rédigé par un individu, ils sont intrinsèquement subjectifs et vont varier en nature, contenu ou granularité. Or un langage de patrons est un construit intellectuel plus sophistiqué, donc encore plus soumis aux biais cognitifs de son rédacteur. Définir un système de valeur explicite c'est se doter de ce que Kawasaki (2004) appelle une *déclaration de mission* : elle permet de prendre en compte et de gérer ces biais. Un système de valeur permet en outre de regarder au-delà de l'apparence comme « Aider l'utilisateur à apprendre le système » pour traduire les idées générales en patrons effectifs pour le processus de conception, expliquant comment les problèmes sont identifiés et résolus d'après certains principes (Mahemoff & Johnston, 1998).

Comme le système de valeur est riche et diffus, faisant appel à un référentiel et des objectifs communs, il est généralement spécifique à une communauté ou à un groupe. La tendance est donc à l'écriture de langages en comités restreints, permettant par ailleurs de cerner plus précisément le sujet à couvrir et donc d'avoir des patrons plus performants (Fincher & Utting, 2002). Les langages de patrons les plus connus sont toujours A Pattern Language et Design Patterns: Elements of Reusable Object-Oriented Software, or ils ont justement été écrits par des petits groupes de gens (quatre personnes) partageant un même état d'esprit. Dearden & Finlay (2006) affirment que le système de valeur intervient concrètement à plusieurs niveaux dans un langage de patrons, en suggérant :

- Les propriétés clés à examiner lors de la recherche des bonnes conceptions susceptibles de fournir des patrons,
- La sélection et le raisonnement fournis pour chaque patron,

- Les processus par lesquels les patrons sont développés et enregistrés,
- La manière dont les patrons sont utilisés.
- Le *principe structurant* est l'organisation des patrons relativement les uns par rapport aux autres. Cette organisation est différente de liens hypertextes conventionnels car il y a causalité : une solution de haut niveau ne peut pas être prédite par la somme de ses composants, elle les dépasse (Salingaros, 2000). Le principe structurant indique quel patron est relié à quel autre, et quelle est la nature de cette relation : parent, enfant, voisin, compétiteur, exclusif, etc. C'est la modélisation mentale que fait une personne, ou un petit groupe de personnes, d'une problématique donnée.

Fincher & Windsor (2000) fournissent de bonnes pistes pour l'élaboration d'un principe structurant, qui devrait au minimum *taxonomiser* (permettre à l'utilisateur de sélectionner et localiser du matériel dans un grand corpus), *approcher* (fournir des patrons plus grands et plus petits, reflétant le comportement « d'incursion opportuniste dans les détails de conception » du concepteur), *évaluer* (supporter différents points de vue, pour changer d'approche ou valider celle qui a été choisie) et *générer* (construire des solutions nouvelles non considérées précédemment).

- Le *formalisme de présentation* permet de capturer les observations de terrain en document compréhensible pour autrui car structuré et explicite. En vertu de leur nature même, les patrons sont souvent formalisés par un gabarit comprenant au moins des entrées pour le problème, le contexte et la solution, ce qui encourage les auteurs de patrons à prendre en compte toutes les composantes de la situation et ne pas se contenter de considérations triviales. Ce gabarit de patron est nommé *schéma*, il est d'ordinaire constant dans une bibliothèque et propre à celle-ci.

Les *modes de présentation* des patrons ne sont pas mentionnés nommément, mais participent de ces deux dernières qualité : ils sont à la fois une incarnation du *principe structurant*, une représentation physique de l'organisation du langage, et une extrapolation de la notion de *schéma de patron* vers un niveau d'abstraction supérieur.

- Les patrons ne documentent pas ce qui est commun mais ce qui est *remarquable*, pour la simple raison que le tout-venant ne nécessitant pas d'explication détaillée est capturé à

moindre frais par des artefacts plus simples ou plus denses en informations, comme les lignes directrices ou les composants précédemment cités. Alexander a par exemple écrit un patron astucieux *Lumière de deux côtés de chaque chambre* plutôt qu'un patron évident *Mettre des fenêtres dans une chambre*.

- La *compréhension* capturée dans le patron est double : elle concerne une modélisation aiguë du problème, expliquant son origine par un contexte précis, et expose le raisonnement qui rend la solution pertinente.
- Les langages sont *génératifs* dans le sens où ils guident un concepteur depuis un problème d'origine vers tous les patrons pertinents, voisins ou enfants, jusqu'à générer une conception complète. Alexander (1979) explique qu'un « langage de patron donne à chaque personne qui l'utilise le pouvoir de créer une variété infinie de bâtiments nouveaux et uniques, comme son langage ordinaire lui donne le pouvoir de créer une variété infinie de phrases ». La générativité d'un langage est sa qualité la plus importante.
- Enfin, les patrons formant le langage sont *communicables* : a-théoriques car écrits avec un vocabulaire commun, agréables car illustrés par des exemples et cohérents entre eux grâce au système de valeur. Des noms bien choisis peuvent servir de synecdoque, d'abréviation pour le patron tout entier, facilitant leur usage.

1.1.3.3 Les bibliothèques de patrons

Devant la difficulté de mise en place des éléments précédents, les praticiens préfèrent la plupart du temps rédiger des *bibliothèques* de patrons (Tidwell, 2007) : un ensemble de patrons liés arbitrairement les uns aux autres et catégorisés de manière rudimentaire. Ils profitent ainsi de la plupart des apports d'un langage de patron pour un coût initial bien plus faible. D'après Salingaros (2000), de simples connexions horizontales suffisent à augmenter drastiquement l'utilité des patrons concernés.

1.1.3.4 Avantages à utiliser des patrons

L'usage des patrons de conception comme outil de gestion de la connaissance apporte de nombreux bénéfices aux équipes de développement des IHM en vertu de leur richesse structurelle.

Premièrement, les patrons de conception provoquent un *effet eureka* quand ils décrètent que la solution n'est plus une prescription *ex machina* flottant dans le vide mais est associée à un problème et un contexte précis. Lorsque le rédacteur prend en compte cette exigence, il se force à dépasser le niveau superficiel du problème pour en tirer, volontairement ou par sérendipité, les éléments fonctionnels de la solution. Poser un patron par écrit force à articuler sa connaissance, révélant bien souvent des lacunes dans des problématiques qui semblaient acquises.

Deuxièmement, « avoir un bon langage de patrons à sa disposition est comme avoir une large équipe d'experts à son côté lors du développement » comme l'explique Booch (2003). Le langage fournit un panel de solutions aidant le concepteur à choisir un paradigme pour attaquer le problème sans avoir à y réfléchir empiriquement chaque fois.

Troisièmement, si le langage de patrons est destiné à une équipe plutôt qu'à un individu, alors il va contribuer à y *construire un consensus*. Il a été expliqué que rédiger patron était subjectif, et que rédiger un langage nécessitait d'explicitier son modèle mental. Cela signifie que les hypothèses et convictions latentes de chacun vont être mises à découvert, confrontées et harmonisées lors de l'élaboration du langage. Le consensus qui en résulte a notamment l'avantage de documenter les décisions déjà prises, facilitant l'intégration de nouveaux membres dans l'équipe, évitant aux équipes distribuées de perdre du temps et posant les fondations des projets ultérieurs.

Quatrièmement, les patrons de conception font office de *langue franche*. Comme les patrons nomment et décrivent des triplets problème-contexte-solution ils établissent un vocabulaire. Comme ils sont aisément *communicables*, ce vocabulaire est accessible à toutes les parties prenantes du processus de conception : techniciens, commerciaux, gestionnaires d'équipe ou de projet, utilisateurs finaux (Erickson, 2000; Reffell, 2006). Les patrons forment un référentiel commun, permettant la participation d'experts de toutes disciplines ou au contraire des utilisateurs finaux⁴, comme un employé de CAE l'a illustré en parlant des patrons de programmation orientée-objet : « Tout le code n'a pas besoin d'être fait par des ingénieurs

⁴ Pour une fois réellement partie prenante, *a contrario* des méthodes « centrées utilisateur » traditionnelles où les utilisateurs sont au final des témoins passifs.

systèmes, il peut être réalisé par des techniciens ou des ingénieurs logiciels. Ils se parlent avec les patrons ». Les patrons élargissent la bande passante de la communication entre interlocuteurs.

Cinquièmement, les patrons de conception sont un outil pédagogique (Scott, 2006). Ils contextualisent la connaissance et la rendent accessible à tous, jouant le rôle d'un outil de vulgarisation. La diffusion du langage a un effet d'immersion pour ses utilisateurs qui, en baignant dans ce corpus hiérarchisé de bonnes pratiques peuvent alors atteindre « la qualité qui ne peut pas être fabriquée mais seulement générée, indirectement, par [leurs] actions » (Alexander, 1979).

À l'usage, ces avantages peuvent se traduire par une influence positive spectaculaire sur le développement logiciel (Beck et al., 1996), améliorant le processus de conception et la qualité des interfaces utilisateur (Kruschitz & Hitz, 2010). Ces bénéfices ont été remarqués par des poids lourds du secteur informatique comme Yahoo! (Leacock, Malone, & Wheeler, 2005), Google (Nesladek, Bauer, Fulcher, Robertson & Palmer, 2010) ou la BBC (British Broadcasting Corporation [BBC], 2010) qui ont tous mis en place une bibliothèque de patrons de conception d'interfaces utilisateur.

1.1.4 Comparatif

Le tableau 1-3 récapitule les caractéristiques des trois artefacts de gestion de la connaissance qui ont été présentés dans ce chapitre.

Chaque artefact adresse des besoins, des intervenants et des étapes du processus de conception logiciel différents ; tous sont donc nécessaires et complémentaires. En outre chacun de ces artefacts, pris séparément mais a fortiori en groupe, améliore la productivité et la qualité :

- La productivité par la réutilisation de solutions prêtes-à-l'emploi plutôt que la conception de zéro à chaque fois.
- La qualité par l'emploi de solutions éprouvées et cohérentes entre elles.

Les patrons de conception semblent être la réponse la plus adaptée pour aider à concevoir les interfaces multiformes (car variées, sophistiquées et très spécifiques) de la gamme de produits de CAE. Dans la suite du mémoire, nous nous concentrerons sur les patrons de conception à

l'exclusion des autres artefacts de gestion de la connaissance et nous garderons à l'esprit la définition que Daniel Sinnig (2004) en a donnée :

« Dans le contexte du développement d'interfaces utilisateur, un patron est une description formalisée d'un concept éprouvé qui exprime une solution non triviale à un problème de conception d'interface utilisateur ».

Tableau 1-3: Comparaison entre lignes directrices, composants et patrons de conception

Critère	Ligne directrice	Composant	Patron de conception
Objectif	Garantir l'homogénéité	Aider la programmation	Aider la conception
Utilisateurs	Ergonomes	Développeurs	Ergonomes
Entité réutilisée	Convention	Implémentation	Conception
Nature de la solution	Conceptuelle	Logicielle	Conceptuelle
Forme	Texte libre + visuels	Code formalisé	Texte formalisé + visuels
Granularité	Spécifique à un système en particulier	Spécifique à un système en particulier	Applicable à plusieurs contextes
Spécificité	Moyenne	Forte	Faible
Durabilité	Moyenne	Faible	Forte

1.2 Les bibliothèques de patrons de conception d'interfaces utilisateur

Dans cette section, nous identifions les bibliothèques de patrons de conception d'interfaces utilisateur les plus populaires et leurs caractéristiques. Nous observons également l'état des connaissances sur des thèmes comme la rédaction ou la gestion technique des patrons : ils peuvent fournir des éléments de compréhension intéressants même s'ils sont périphériques de notre sujet principal, la comparaison de bibliothèques.

1.2.1 Méthodologie

Il existe une multitude de bibliothèques de patrons de conception d'interfaces utilisateur (une recherche Google pour "*UI design pattern library*" retourne 54,800 résultats) ; notre revue de littérature se concentre sur celles qui répondent aux critères suivants :

- Le sujet couvert par la bibliothèque doit être en adéquation avec les besoins de CAE, à savoir le développement d'applications pour PC ou, dans une moindre mesure, d'applications Web riches,
- La bibliothèque doit avoir une bonne validité apparente, c'est-à-dire avoir été citée par les auteurs mentionnés dans la bibliographie ou ne pas lister moins de 10 patrons. Cela exclut de nombreux efforts sur le Web s'apparentant plus à des collections de composants qu'à des bibliothèques de patrons.
- Le ou les modes de présentation correspondent à la manière dont les patrons d'une bibliothèque sont affichés au lecteur. Quelques bibliothèques disponibles sous forme de site Web offrent plusieurs modes de présentation, généralement fournis automatiquement par le système de gestion de contenu sur lequel elles sont bâties : patrons les plus commentés, patrons les plus populaires, patrons récents, brouillons... Nous définirons comme *principal* le mode de présentation affiché de manière proéminente (en premier, en plus gros, sur la plupart des pages) et permettant de naviguer dans toute la bibliothèque. Pour les bibliothèques sous forme de livre, nous avons de la même manière considéré les énumérations de patrons commentées par l'auteur comme mode de présentation principal, plutôt que les tables des matières de ces ouvrages. La revue de littérature compare les modes de présentation principaux des bibliothèques.
- Les méta-données parfois associées au patron (auteur, date, licence juridique, etc.) qui n'apportent pas de contenu effectif pour aider la conception ou qui sont minimisées à l'écran ne sont pas mentionnées dans les schémas de patrons.
- Les sujets couverts par la bibliothèque ne sont pas toujours explicitement mentionnés par leurs auteurs. Dans ces cas, signalés par la mention (*présumé*), nous avons essayé d'identifier nous-mêmes les thèmes sur lesquels la bibliothèque portait.

1.2.2 Résultats

Les résultats de la recherche et de la caractérisation des bibliothèques de patrons de conception d'interfaces utilisateur sont disponibles, avec des exemples de patrons, dans l'annexe A.

1.2.3 Analyse

Une étude attentive des résultats précédents nous permet de dégager plusieurs éléments.

1.2.3.1 L'absence d'un schéma standard et d'une notation formelle

Il n'existe pas de schéma standard de patron, chaque bibliothèque utilisant un formalisme qui lui est propre (annexe A.5). Les grandes lignes sont cependant identiques et certains éléments reviennent fréquemment : *titre, problème, contexte, solution et exemple*.

Par ailleurs, tous les patrons sont rédigés en texte simple, sans balisage particulier qui les rendrait intelligibles pour l'ordinateur, exception faite de ceux de Welie (2000).

1.2.3.2 De la pertinence d'une bibliothèque spécifique

Beaucoup de bibliothèques concernent principalement le Web, même quand elles se présentent sous des appellations plus génériques comme « Interaction Design Patterns » ou « UI Design Patterns » (annexe A.2). Elles ignorent les spécificités techniques, contextuelles ou culturelles d'autres plates-formes comme les téléphones (intelligents ou pas), les assistants numériques personnels et surtout les systèmes d'exploitation de bureau. C'est un double inconvénient par rapport aux besoins de CAE : d'une part les solutions proposées dans ces bibliothèques concernent plutôt des sites *brochureware* (Taleb, Seffah & Abran, 2007) grand public et ne sont que partiellement applicables à des systèmes temps-réels militaires, d'autre part le Web est un sujet très large, alors que les gammes de produits CAE sont extrêmement ciblées et pourraient profiter d'un langage taillé sur mesure comme présenté plus haut. Mahemoff & Johnston (2001) pèsent en ce sens :

« Beaucoup de collections de patrons ont une grande envergure, et nous soutenons que cela résulte en des patrons qui ne sont pas aussi interdépendants que ce qu'une envergure plus étroite permettrait », (traduction libre du texte de la page 1).

Cet état de fait pèse en faveur de la rédaction d'une bibliothèque de patrons spécifique à CAE, s'inspirant de l'esprit, de la forme et du contenu des bibliothèques existantes mais sans être

dérivée d'une d'entre elles en particulier. Ainsi, le schéma des patrons CAE pourrait accommoder le besoin singulier de l'entreprise (servir les développeurs logiciels en plus des experts en facteurs humains) avec des entrées spécifiques, comme par exemple des liens entre un patron et différents dépôts de code permettant de l'instancier. Vlissides (1997) justifie la variété des schémas de patron :

« Clairement, un format de patron ne convient pas à tout le monde. Ce qui convient à tout le monde est le concept général d'un patron comme vecteur de capture et de transport de l'expertise » (traduction libre du texte).

Welie (2006) reprend cet argument en détail et y adjoint des pistes de rédaction dont nous aurons ultérieurement l'usage :

« Dans la communauté des patrons, beaucoup d'attention a été portée au format des patrons - quelles sections ils sont supposés avoir, comment ces sections sont nommées, etc. J'ai remarqué que cela compte moins que ce qu'on pense, tant que les réponses basiques sont là. (...) Rappelez-vous que vous écrivez ce patron pour un public humain. Il doit être lisible, et trop de sections - ou trop de jargon, qui peut expliquer ce que +Forces+ veut dire? - empêche le lecteur de déchiffrer ce que vous êtes en train de dire. (...) Jennifer et moi avons chacun pris un chemin arbitraire et changé nos formats [de patrons] plusieurs fois, à chaque fois que cela semblait approprié » (traduction libre du texte).

1.2.3.3 Des bibliothèques *sur* des sujets, pas *pour* des organisations

Il existe une séparation claire entre les bibliothèques conçues sous forme de livres et celles disponibles sur le Web, quels que soient les sujets qu'elles traitent. Les bibliothèques sous forme de livre contiennent généralement plus de patrons que celles disponibles sur le Web, et leurs patrons sont plus riches : plus longs, mieux rédigés et s'appliquant à plus de plates-formes, sans pour autant perdre leur aspect pratique. Les patrons sont livrés avec beaucoup plus de contexte et de conseil de la part de l'auteur ; ils font généralement partie d'une réflexion de plus grande ampleur, expliquée sur plusieurs pages là où les bibliothèques disponibles sur le Web se contentent de quelques mots pour introduire leurs patrons. Les patrons des bibliothèques sur le Web sont plus structurés, car découpés en de nombreuses sections cloisonnées et titrées, bien qu'ils aient moins de contenu ; il en résulte une grande lisibilité. Inversement les patrons dans la littérature tendent à être rédigés comme des blocs uniques et denses, amenant plus d'information au prix de la facilité de lecture. Au final, les patrons reflètent la nature des media dans lesquels ils sont stockés : consommation rapide mais superficielle pour le Web, ardue mais exhaustive pour les livres.

Dans tous les cas, il convient de noter que ces bibliothèques ne sont pas rédigées pour des organisations spécifiques, mais pour couvrir des aspects plus ou moins particuliers de la discipline de la conception des IHM, comme le précisent leurs auteurs respectifs. Nous n'avons pas identifié de bibliothèque de patrons mise en place expressément pour une entreprise, ou prenant des mesures particulières pour les développeurs logiciels (rédaction, mise à disposition, incitation, support...).

1.2.3.4 Un mode de présentation uniforme pour un public expert

Nous observons dans que toutes les bibliothèques utilisent un mode de présentation unique et récurrent (annexe A.5), avec quelques menues variations : un index alphabétique, segmenté en groupes d'une dizaine de patrons, chaque entrée de l'index recevant éventuellement une ligne de commentaire, comme l'illustre la figure 1-4 tirée de Tidwell (2005). La segmentation suit généralement un modèle mental et un vocabulaire d'ergonome comme par exemple chez Duyne (2007) : *Creating a navigation framework* ou *Building trust & Credibility*.

Dans la suite du mémoire, ce mode de présentation alphabétique sera dit *de référence*. Il est pertinent de le corréliser avec le public explicitement visé par ces bibliothèques (annexe A.2) : concepteurs de toutes sortes (Web, expérience utilisateur ou interfaces utilisateur), mais pas développeurs logiciels. La section suivante examine ce point en détail.

Coram & Lee (1998) ajoutent à ce mode de représentation textuel une représentation de l'organisation de leur bibliothèque et des relations entre ses patrons sous forme de diagramme. Nous relèverons que la seule bibliothèque à être représentée de manière graphique, et incidemment que c'est une des plus petites.

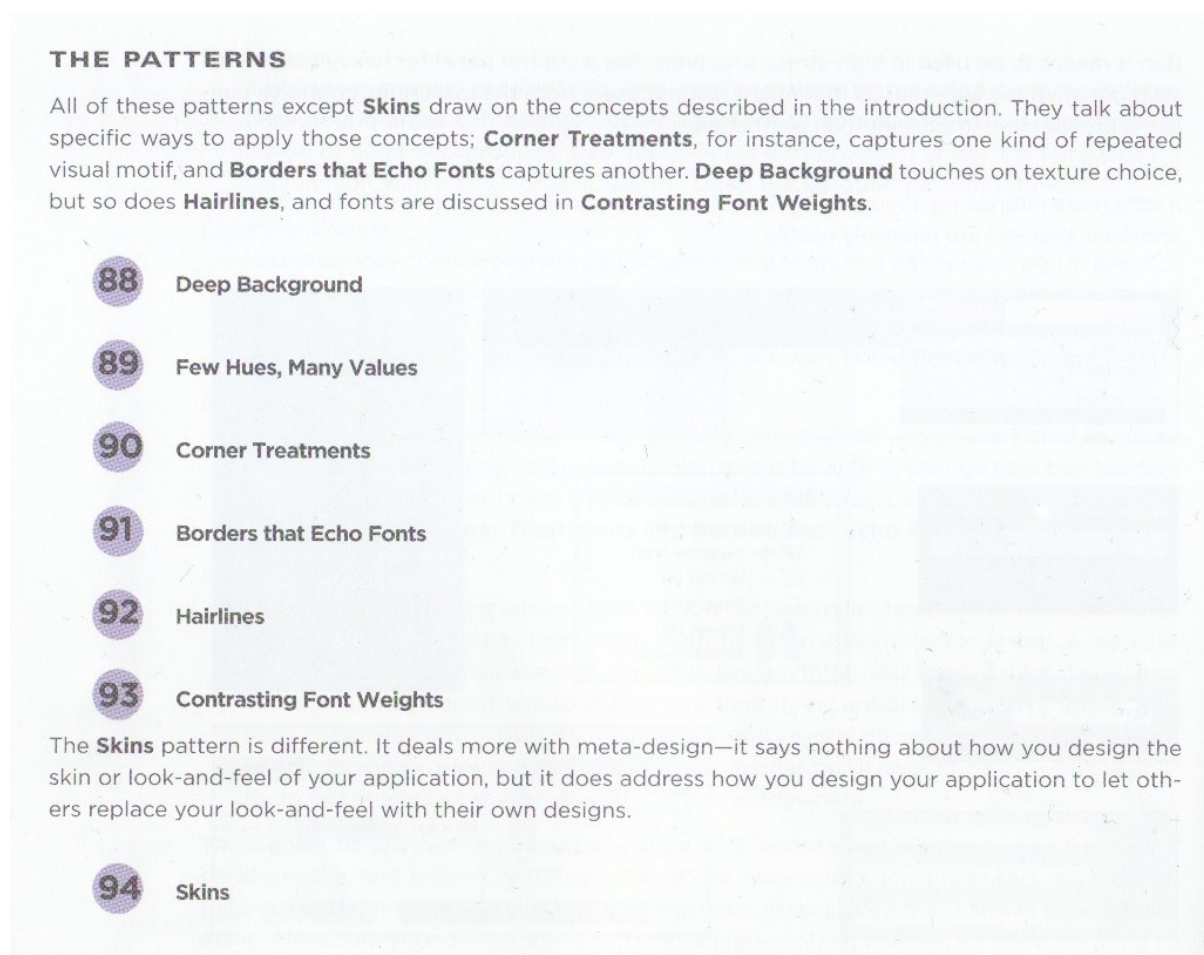


Figure 1-4: Mode de présentation des patrons du chapitre *Making it look good*

Nous en déduisons qu'en ciblant un public expert en IHM, les auteurs de ces bibliothèques ont fait un compromis en faveur de l'efficacité, fournissant des listes denses, d'un seul tenant et rapides à balayer pour ces utilisateurs avancés. La réciproque d'un tel mode de présentation est l'absence de contexte, nuisant à l'incitation ou au guidage pour des utilisateurs moins avancés. D'après Kruschitz & Hitz (2010), des schémas de catégorisation multiples sont pourtant demandés par tous les types d'utilisateur des bibliothèques :

« Les utilisateurs veulent avoir plusieurs structures hiérarchiques pour trouver des patrons dans une collection. Il y a un besoin de schémas d'organisation et de catégorisation (...) pour différents problèmes et situations » (traduction libre du texte).

1.2.3.5 Autres éléments notables : les systèmes de gestion de contenu, les relations entre patrons, la découverte des patrons

La même étude de Kruschitz & Hitz (2010) révèle que 71,35% des utilisateurs de patrons n'utilisent aucun outil de gestion de patrons et stockent leur patron ad-hoc dans des wikis personnels et surtout dans des documents bureautiques, ne leur permettant pas d'avoir les structures hiérarchiques multiples qu'ils voudraient pour trouver des patrons dans une bibliothèque. Pourtant, un environnement Web serait tout désigné pour héberger un langage de patrons puisqu'ils partagent la même nature : un réseau d'éléments reliés par des hyperliens (Welie & van der Veer, 2003).

La mise en place d'un tel système de gestion de contenu pour héberger la bibliothèque de patrons de CAE est détaillée dans la section 2.4.3 *Développement du système de gestion de contenu*.

Attardons-nous à présent sur les relations entre patrons. Une minorité de bibliothèques (quatre sur 11), indépendamment de leur support papier ou Web, ne relie pas ses patrons les uns aux autres : chaque patron y est une entité finie. La majeure partie des bibliothèques établissent des relations entre leurs patrons, sous la forme de liens hypertextes (Web) ou de pointeurs (livres). Ces liens sont en tous points identiques à des liens hypertextes ordinaires.

Janeiro, Barbosa, Springer & Schill (2010b) soulignent l'importance des relations entre patrons dans un langage, et avancent qu'elles pourraient aider les développeurs logiciels à mieux se servir des langages de patrons. Ils proposent à cet égard l'emploi de sept relations typées explicitant la nature des liens entre patrons : *Utilisé avec*, *Équivalent*, *Similaire*, *Équivalent sous conditions*, *Réalisation*, *Amélioration* ou *Conflit*. Grâce à ce suivi à la trace de l'information, les développeurs logiciels parcourant un patron donné n'ont pas besoin de lire les patrons liés en intégralité pour déterminer s'ils sont intéressants par rapport à leur problème.

Son inconvénient est de supporter les développeurs qui ont déjà trouvé un patron, alors que la présente étude se concentre sur la manière de fournir un premier patron à un public néophyte. Autrement dit c'est une approche très intéressante, qui augmente la satisfaction des utilisateurs d'environ 20% d'après JaneiroBarbosa, Springer & Schill (2010a), mais elle sera plutôt appliquée dans un deuxième temps : nous nous contenterons d'abord de relier les patrons CAE entre eux par des liens hypertextes ordinaires.

Enfin, les activités de découverte et de rédaction des bibliothèques de patrons ne sont que très peu mentionnées dans les ouvrages revus. Nous avons tout de même trouvé deux processus formels et complets pour la rédaction de patrons de conception à l'unité : le processus des 7C (tableau 1-4) et le Flux des tâches pour la création de patron (tableau 1-5).

Il ne serait pas judicieux d'employer ces processus chez CAE.

Le premier concerne en effet des bibliothèques publiques, piochant des patrons dans d'autres bibliothèques et soumettant ses propres patrons à l'approbation de la communauté des ergonomes. Ceci n'est pas possible pour une entreprise comme CAE travaillant dans le secteur de la défense. Le second est destiné aux petites et moyennes entreprises, dans lesquelles le rédacteur de la bibliothèque est en contact direct avec ses utilisateurs et peut leur demander plusieurs boucles de rétroaction, ce qui n'est pas non plus le cas chez CAE : l'entreprise est une multinationale dont les effectifs sont distribués organisationnellement, géographiquement et temporellement.

La rédaction des patrons pour logiciels de contrôles de simulateurs de vol chez CAE suivra donc autant que possible un processus inspiré de l'existant, mais devra s'en écarter parfois en raison du secteur d'activité particulier de l'entreprise et du public prévu pour sa bibliothèque de patrons.

Tableau 1-4 : Le processus des 7C

Texte librement traduit de Gaffar (2005)

Collecter	Placer différents travaux de recherche sur les patrons dans un seul dépôt central de données.
Conformer	Transcrire les différents formats en un style.
Certifier	Déterminer un domaine et une terminologie claire.
Contribuer	Recevoir l'avis de la communauté des patrons.
Connecter	Établir des relations sémantiques entre patrons via un modèle relationnel.
Catégoriser	Définir des catégories claires pour les patrons qui les relient aux processus d'assimilation.
Contrôler	Employer un format lisible par la machine pour les outils à venir.

Tableau 1-5 : Flux des tâches pour la création de patrons

Texte librement traduit de Lammi (2007)

1.	Identifier et suggérer un patron
2.	Évaluer et commenter les suggestions de nouveaux patrons
3.	Identifier et assigner un auteur
4.	Écrire et soumettre les patrons
5.	Réviser et publier
6.	Évaluer et commenter les patrons publiés
7.	Mettre-à-jour et republier les patrons

1.3 Les difficultés des développeurs logiciels

« Les patrons communiquent des idées de conception d'un ergonome à un autre ; les ergonomes seront probablement toujours les premiers auteurs et le premier public des patrons de conception d'interface utilisateur » Traduction libre du texte de Reffell (2006).

Ne pas prévoir explicitement les patrons de conception pour les développeurs logiciels les empêche de s'en servir et fait des patrons l'apanage involontaire mais effectif des ergonomes. Cette exclusivité de fait peut expliquer que les patrons de conception n'aient pas atteint l'ampleur et l'acceptation envisagées par les pionniers du domaine, en dépit de leur potentiel évident et reconnu Seffah (2010).

Cette section est un tour d'horizon des difficultés qu'ont les développeurs logiciels à se servir des bibliothèques des patrons de conception d'interfaces utilisateur. Nous nous sommes basés sur le travail de Breiner et al., (2010) pour isoler quatre facteurs responsables des difficultés que les développeurs ont avec les bibliothèques de patrons actuelles. Il s'agit de la *variabilité des schémas de patrons*, de l'*absence de guidage dans les bibliothèques de patrons*, de l'*absence de conseil lors de la lecture d'un patron* et du *manque d'outils informatiques*.

1.3.1 Pas de guidage dans les bibliothèques de patrons

La spécification d'un produit (logiciel ou autre) ne se fait pas par une énumération de problèmes à résoudre dans un contexte mais via une liste de fonctions à accomplir dans une enveloppe de performance donnée : il n'y a *a priori* pas de lien naturel entre un produit donné et les patrons de conception qui pourraient s'y appliquer.

L'aptitude des développeurs logiciels à trouver le bon patron dans un large corpus est donc primordiale⁵, d'autant plus que pour résoudre un problème donné les développeurs ne doivent pas chercher qu'un patron mais tout un sous-ensemble de patrons. La revue de littérature montre que les catégorisations des patrons varient autant que leurs schémas mais les modes de présentations qui en découlent se ressemblent tous dans leur sobriété (annexes A.3 et A.5).

Faute de mode de présentation évident, facile à vulgariser et à apprendre, les développeurs logiciels ont du mal à se représenter l'ampleur et l'organisation des bibliothèques ; ils y naviguent donc à tâtons. Même s'ils arrivent dans la bonne section, l'absence de guidage pousse les développeurs confrontés à une liste de liens obscurs à procéder par essai-et-erreur, faute de mieux. La faible satisfaction qui en résulte ne les incite pas à utiliser des patrons de conception.

En conséquence, les développeurs ont du mal à identifier les patrons pertinents par rapport à leur problème, et quels patrons peuvent être utilisés ensemble. Ne pas réussir à identifier les patrons pertinents pour un problème donné est un véritable déboire :

- Appliquer un patron inapproprié peut conduire à une dégradation de l'interface utilisateur. Par exemple il ne serait peut-être pas judicieux de permettre une *Annulation à plusieurs niveaux*⁶ dans une transaction bancaire en cours car cela pourrait laisser l'utilisateur confus quant à l'état de son patrimoine.
- Appliquer un patron générique n'apporte pas beaucoup de valeur à la conception et gaspille des ressources qui seraient mieux investies ailleurs. Des *Couleurs de rangs*

⁵ En admettant qu'ils se soient arrêtés à une seule bibliothèque. Les développeurs peuvent tout simplement se perdre quand ils doivent utiliser plusieurs bibliothèques en même temps (Gaffar et al., 2005).

⁶ *Multi-level undo*, tiré de la bibliothèque Designing Interfaces de Tidwell (2005).

*alternées*⁷ sont toujours bonnes à avoir dans une liste large mais n'apportent pas la plus grande contribution à l'utilisabilité globale de l'application.

Notons que la généricité, ou réciproquement, la pertinence d'un patron de conception pour une interface en particulier peut être difficile à mesurer en raison de la nature « ouverte à l'interprétation » des patrons de conception. Pour la suite du mémoire, nous définirons la pertinence d'un patron relativement à un écran comme le produit de l'applicabilité du patron à cet écran (« Est-il techniquement possible d'utiliser le patron sur cet écran ? ») et de la spécificité intrinsèque du patron. Nous verrons un peu plus tard comment définir objectivement ces deux dimensions.

Gaffar (2005) avance en plus que ces problèmes sont amplifiés par les phénomènes *d'échelle* (la difficulté augmente avec le nombre de patrons) et de *visibilité* (les nouveaux patrons sont noyés dans la masse) qui font qu'au-delà de 50 patrons il n'est plus possible de les organiser linéairement (Beck et al., 1996).

Au final, les utilisateurs passent entre deux et 10 minutes à chercher des patrons (Lammi, 2007) dans une bibliothèque, mais ne parviennent pas toujours à en trouver dans un temps aussi court (Kruschitz & Hitz, 2010) et en éprouvent de la frustration (Blackwell & Fincher, 2010). Gaffar (2005) résume la situation en dénonçant l'idée que la découverte de bons patrons soit simple comme la « première illusion » sur les patrons de conception.

1.3.2 Pas de conseil lors de la lecture d'un patron

Même une fois le ou les bons patrons trouvés, leur utilisation par les développeurs logiciels est problématique. La compréhension du patron, la capacité à le transcrire en une solution effective et pertinente à un problème et à un contexte donné (*assimilation*), à en dévier au besoin, ou à le combiner avec d'autres patrons en une solution holistique sont autant de notions connues et implicites pour les ergonomes mais difficiles à acquérir pour les développeurs logiciels (Breiner et al., 2010).

Ces derniers doivent en plus se soucier de l'instanciation des patrons trouvés (leur traduction en code informatique). Cette étape pourrait sembler secondaire aux ergonomes, dont le livrable

⁷ *Row striping*, tiré de la bibliothèque Designing Interfaces de Tidwell (2005).

principal est la conception, mais c'est en fait une activité cruciale puisqu'elle va faire le lien avec le développement logiciel en lui livrant un artefact sous forme de code informatique. Cette activité, du ressort des développeurs logiciels, n'est pas prise en compte par les bibliothèques de patrons existantes.

1.3.3 La variabilité des schémas de patron

Il n'y a pas de consensus sur une description standard des patrons de conception d'interfaces utilisateur (Janeiro et al., 2010a). Chaque bibliothèque utilise sa propre structure descriptive pour ses patrons, avec une terminologie, une classification (via son principe structurant) et des objectifs différents. Même si la plupart des patrons reprennent des éléments communs, ils sont traités et présentés dans des contextes différents par leurs auteurs. Il en résulte un domaine « décourageant pour les néophytes, qui peuvent trouver difficile de démêler les caractéristiques conceptuelles de l'approche et à plus forte mesure sa contribution » (Dearden & Finlay, 2006).

1.3.4 Le manque d'outils informatiques

Les variations de schéma et d'organisation précédemment exposées entravent la prédictabilité, la répétabilité et l'uniformité requises pour un traitement informatique des patrons. Le support papier de certaines bibliothèques complique encore l'automatisation : non seulement ce medium est intrinsèquement moins utilisable par un ordinateur, mais en plus ces patrons sont généralement rédigés de manière plus dense, peu structurée et peu formalisée, et sont donc encore plus ardues à interpréter informatiquement.

La difficulté de lecture des patrons par un ordinateur explique « qu'il n'y ait pas d'outil de support pour la conception par patrons d'interfaces utilisateur » (Seffah & Javahery, 2002). L'utilisation de patrons dans un processus de conception reste donc une tâche manuelle, relativement lente et difficile, alors que certaines opérations récurrentes (représentant une charge de travail importante) pourraient être automatisées. Par exemple :

- La recherche, le filtrage ou le tri de patrons oscille entre rudimentaire et impossible, exigeant souvent de connaître l'intégralité du contenu de la bibliothèque pour trouver le patron adéquat.

- De même l'utilisation générative des patrons, c'est-à-dire la fourniture de patrons spécifiques une fois un premier patron choisi, se fait sans assistance. En l'absence d'interdiction automatique de patrons, l'utilisateur doit lire et comprendre les patrons reliés en intégralité pour déterminer s'ils sont applicables, avant même de savoir s'ils sont pertinents. Il en résulte une grande perte de temps. De même il n'a y pas de suggestion automatique de patrons pour compléter ou continuer la conception en cours, etc.
- La gestion et l'échange de patrons entre bibliothèque, le stockage des patrons dans une organisation, leur mise en relation ou tout autre opération de gestion de contenu est clairement limitée et fastidieuse (Breiner et al., 2010).
- Et enfin l'assimilation et l'instanciation de patrons en composants spécifiques à une fonction et à une plate-forme sont pour le moment manuelles. Il n'y a pas de génération de code automatique.

1.3.5 Conclusion

Les variations dans la rédaction et la catégorisation des patrons de conception d'interfaces utilisateur sont des obstacles rédhibitoires à l'utilisation des bibliothèques *per se*, mais qui en plus empêchent la mise en place d'outils informatiques pour y remédier.

Les utilisateurs potentiels des patrons de conception d'interface utilisateur sont confrontés à un paradoxe : pour exploiter une bibliothèque de patrons, ils ont besoin des ressources qu'ils essaient justement d'économiser, à savoir du temps et de l'expertise en ergonomie des IHM. Il leur faut devenir des experts en patrons pour pouvoir trouver et appliquer des patrons (Gaffar, Seffah & Poll, 2005). Les développeurs logiciels, responsables involontaires de tâches de conception d'IHM (Krug, 2005), sont ainsi privés d'un outil qu'ils jugent pourtant hautement bénéfique (Seffah & Javahery, 2002).

1.4 Synthèse de la revue de littérature

Les patrons de conception sont un artefact de gestion de la connaissance populaire et à l'efficacité reconnue, mis en œuvre par les plus grands acteurs du secteur logiciel. Les bibliothèques de patrons de conception d'interfaces utilisateur existantes sont toutefois rédigées exclusivement par

et pour des experts en IHM, rendant leur accès et leur application difficiles pour les développeurs logiciels qui n'ont ni le temps ni l'expertise nécessaire pour en profiter.

Un des aspects les plus ardu pour les développeurs logiciels est la navigation dans une bibliothèque de patrons. En effet, toutes les bibliothèques présentent leurs patrons de manière similaire et peu adaptée aux caractéristiques et aux besoins des développeurs logiciels. Des modes de présentation différents pourraient épargner aux développeurs les étapes difficiles et potentiellement longues de lecture, compréhension et comparaisons de plusieurs patrons.

Nous formalisons cette hypothèse dans le chapitre suivant et y détaillons les activités qui vont nous permettre de l'évaluer.

CHAPITRE 2 PROBLÉMATIQUE, OBJECTIFS ET DÉVELOPPEMENT D'UNE BIBLIOTHÈQUE DE PATRONS POUR INTERFACES UTILISATEUR

Ce chapitre présente la problématique et les hypothèses du mémoire, ainsi que les actions que nous avons menées pour créer la bibliothèque de patrons de conception de CAE. Nous commencerons par détailler les études de faisabilité effectuées, puis nous verrons le recensement des bonnes pratiques au sein de l'entreprise, leur rédaction sous forme de patron et enfin leur mise à disposition sur l'intranet via un système de gestion de contenu dont la mise en place est également détaillée.

2.1 Question et activités de recherche

Il convient de rappeler le contexte et les enjeux de notre étude avant d'en exposer les objectifs. L'interface utilisateur des logiciels de contrôle des outils de simulation que construit CAE est conçue en partie par des développeurs logiciels n'ayant pas reçu de formation en ergonomie. Une bibliothèque de patrons de conception d'interface utilisateurs pourrait les soutenir dans cette tâche en fournissant un répertoire de bonnes pratiques et un support de communication entre les développeurs et les ergonomes, mais il a été établi que les développeurs logiciels avaient du mal à se servir des bibliothèques existantes.

Notre question de recherche est donc :

Comment permettre à des développeurs logiciels non formés en ergonomie d'utiliser des patrons de conception d'interfaces utilisateur ?

L'objectif général du mémoire relativement à cette question est :

Développer des modes de présentation des patrons d'une bibliothèque adaptés aux développeurs logiciels.

Nous allons effectuer les activités suivantes pour atteindre l'objectif général :

1. Recenser les patrons de conception pour logiciels de contrôle de simulateur de vols,
2. Mettre au point un style de rédaction adapté aux développeurs logiciels et rédiger les patrons selon ledit style,

3. Diffuser les patrons sur le réseau intranet de l'entreprise sous la forme d'une bibliothèque de patrons,
4. Mettre au point des modes de présentation permettant aux développeurs logiciels de naviguer dans cette bibliothèque de patrons,
5. Évaluer l'effet des innovations proposées.

Ces activités sont chacune détaillées dans les chapitres suivants.

2.2 Hypothèses

Étant admis que le mode alphabétique plat est celui de référence, nous proposons :

1. L'usage des trois nouveaux modes de présentation améliore significativement la pertinence des patrons trouvés par rapport au mode de présentation de référence.
2. L'usage des trois nouveaux modes de présentation améliore significativement la satisfaction des développeurs par rapport au mode de présentation de référence.

2.2.1 Justification de l'originalité

Les bibliothèques existantes utilisent toutes un mode unique et semblable pour présenter leurs patrons. Ce mode de présentation est un index alphabétique plat, segmenté et décrit avec un vocabulaire d'ergonome. Les patrons y sont listés avec peu ou pas de contexte.

Il n'existe pas de bibliothèque proposant plusieurs modes de présentation de ses patrons, ni des modes prévus explicitement pour les développeurs logiciels.

2.2.2 Réfutabilité

Les hypothèses seront réfutées si les évaluations ne montrent pas de progression significative ou si la progression est trop faible au regard de l'investissement consenti.

2.3 Travaux préparatoires

Cette section décrit quelques études de faisabilité technique et opérationnelle menées au préalable de la rédaction de la bibliothèque de patrons. Il s'agit de travaux de préparations périphériques destinés à dégrossir le sujet en gardant à l'esprit :

- La nécessité du secret inhérente aux secteurs d'activité de CAE (défense ou santé),
- La nécessité de supporter les développeurs logiciels en plus des experts en facteurs humains,
- Les ressources limitées dont dispose l'équipe d'ingénierie des facteurs humains pour mettre en place la bibliothèque.

Nous verrons d'abord la pertinence de faire une nouvelle bibliothèque plutôt que d'en réutiliser une, puis nous évaluerons le besoin de rendre ces patrons lisibles par l'ordinateur et enfin nous nous intéresserons à la manière à présenter au mieux les liens entre patrons.

2.3.1 L'automatisation mise de côté

La notation des patrons de conception de manière formelle en vue de leur traitement informatique est un axe de recherche académique majeur. En effet, saisir les patrons sous forme de code informatique permettrait d'obtenir les avantages précédemment évoqués, mais aussi d'après Gaffar et al. (2005) :

- L'interopérabilité entre plusieurs bibliothèques, chacune pouvant importer ou exporter des patrons sans intervention humaine.
- La multiplication des modes de présentation possibles pour une bibliothèque donnée, l'apparence étant découplée du contenu.

Cet axe de développement n'a pas été activement poursuivi chez CAE.

En effet si l'idée d'un support automatisé aux tâches de conception est une perspective attirante, les outils permettant effectivement de le faire ne sont pas encore disponibles. Pire encore, il n'existe pas de notation standard dans la communauté : certains auteurs (Gaffar et al., 2005), 2005) recommandent du XML générique, d'autres ont déjà publié des bibliothèques en PLML, une variante du XML, comme Welie (2000), ou en JSON, basé sur JavaScript, comme Leacock, Malone, Wheeler & Scott, (2006) ; la plupart, comme Morville & Callender (2010), Crumlish & Malone (2009) ou Behrens (2008), écrivent sans aucun formalisme informatique.

En l'absence de standard, une autre forme de notation a été un temps considérée pour représenter les patrons chez CAE : les étiquettes (plus précisément les *machine-tags*). Leur souplesse technique et leur capacité à supporter des *folksonomies* (ontologies légères ascendantes)

semblaient en adéquation avec les besoins de CAE. Les machine-tags ont cependant été délaissés, pour les mêmes raisons que les autres formes de notation : le manque d'outils et des incertitudes sur leur retour sur investissement (Limpens, Gandon & Buffa, 2009; Mika, Atserias, Ciaramita & Zaragoz, 2008).

De même la génération automatique de code n'est pas à l'ordre du jour. La bibliothèque de patrons de CAE est encore partiellement en cours de rédaction (annexe C.1) et donc soumise à des modifications fréquentes. Le supplément d'effort nécessaire au balisage des patrons pour qu'ils soient interprétables par l'ordinateur serait considérable, pour un bénéfice que Vlissides (1997) juge faible :

« Je peux affirmer avec certitude que le bénéfice des patrons vient essentiellement de leur application en tant que tel, c'est-à-dire sans soutien d'aucune sorte. (...) Il y aura peut-être des bénéfices latents dans un soutien méthodologique ou automatique, mais je suis convaincu que ce sera la cerise sur le gâteau, pas le gâteau lui-même ni même un de ses étages » (traduction libre du texte).

Enfin, l'interopérabilité n'est pas nécessaire à la bibliothèque de patrons de CAE. L'import de patrons présente un intérêt limité en raison de l'écart entre les sujets couverts par la plupart des bibliothèques (Web, réseaux sociaux, terminaux mobiles) et les produits conçus par CAE (systèmes temps réel complexes). L'export de patrons est complètement proscrit pour préserver tant la compétitivité de l'entreprise que les secrets liés au secteur de la défense.

Au final, les patrons CAE sont pour le moment rédigés sans effort particulier pour les rendre lisibles par la machine. C'est une position acceptable, comme l'exprime Tidwell (2006) :

« Vous n'écrivez pas du code non plus. Ou des spécifications, ou un catalogue de composants. Vous trouverez peut-être personnellement utile de déterminer une logique très précise pour vos patrons, mais je vous garantis que la plupart des lecteurs s'en ficheront. Cela veut aussi dire que je ne pense pas que les patrons d'interface utilisateurs, tels que je les écris, aient beaucoup de valeur pour l'automatisation de la conception ou la génération de code » (traduction libre de l'auteur).

L'automatisation des patrons de conception demeure tout de même un sujet prometteur à explorer pour des bibliothèques plus stables. Des paradigmes de notation plus légers comme les microformats (Berriman et al., 2011) ou les microdonnées (Hickson, 2011) permettraient peut-être de transformer les patrons en *données intelligentes* (Gaffar, 2005) lisibles par l'ordinateur.

2.3.2 Des relations entre patrons *a minima*

Nous avons déjà écarté l'idée de relations sémantiquement typées pour les patrons de CAE. Nous allons maintenant considérer une autre piste de recherche récurrente sur les relations entre patrons, leur représentation graphique.

La bibliothèque pourrait a priori être visualisée sous forme visuelle plutôt que hypertextuelle, explicitant les relations et groupements entre patrons. En pratique, l'exécution d'un tel diagramme s'est révélée ardue : le nombre important de relations croisées entre patrons rend la lecture difficile, à moins que le diagramme n'ait une taille gigantesque dépassant de loin celle d'un moniteur commun. De plus le diagramme ne permet pas le dévoilement progressif, il montre directement toute l'information ; les utilisateurs ne savent pas quoi faire du diagramme ni par où commencer, et en l'absence de guidage et de contexte, finissent par se décourager. L'idée même de représentation graphique a été contestée par les utilisateurs qui ne s'attendaient pas à trouver un contenu d'une telle nature (diagramme) sur le site intranet de l'entreprise.

En somme, les relations entre patrons représentent un autre grand potentiel d'amélioration pour l'utilisabilité des bibliothèques de patrons mais elles sont encore trop mal connues pour que leur exploitation soit intéressante aujourd'hui, d'autant plus dans notre cas où la priorité est d'amener des néophytes à trouver un premier patron, avant d'être en mesure d'exploiter efficacement ses relations.

2.4 Rédaction des patrons

Cette section décrit comment la rédaction des patrons a débuté avec un recensement des besoins et des bonnes pratiques de l'entreprise, suivie d'une rédaction du matériel obtenu selon un formalisme approprié pour le public visé, les développeurs logiciels. La mise en place d'un système de gestion de contenu permettant de partager ces patrons sur le réseau est ensuite détaillée.

2.4.1 Recensement des besoins et des bonnes pratiques

Le recensement des besoins de CAE en matière d'ergonomie des IHM va fournir directement les composantes *problème* et *contexte* de nos patrons, alors que le recensement des bonnes pratiques va nous fournir la composante *solution* des patrons.

Nous avons commencé par nous entretenir avec les deux membres de l'équipe d'ingénierie des facteurs humains de l'entreprise pour avoir une vue d'ensemble du sujet et profiter de leur expertise, notamment à propos des qualités des interfaces des produits CAE, de leurs déficiences récurrentes et des efforts en cours pour y remédier. Nous avons ensuite travaillé avec une trentaine de développeurs logiciels puisque ce seront eux les utilisateurs finaux de la bibliothèque de patrons de CAE ; leurs connaissances implicites ont été relevées en observant les développeurs en action à leur poste de travail avec plusieurs techniques : entrevues individuelles semi-structurées, présentation pas à pas d'un produit ou information à la demande. Après cela nous avons étudié l'organisation dans son ensemble pour voir quels patrons lui seraient profitables. L'analyse des habitudes de travail ou des objectifs d'affaires de CAE, de ses matériels, logiciels, documents de références et autres éléments tel qu'indiqué par l'Organisation internationale de normalisation (ISO, 1998) nous a ainsi permis de cerner les notions capturant les caractères présents et futurs des interfaces des outils de simulation de CAE. Ces notions font office de système de valeur rudimentaire :

- Prévoir pour des utilisateurs experts et des conditions extrêmes,
- Prévoir pour le tactile autant que pour le clavier/souris,
- Limiter au maximum les fenêtres enfants,
- Préférer les interfaces à document unique aux interfaces à documents multiples.

Avec cette vue d'ensemble en tête, nous avons cherché des réponses plus concrètes. Les problèmes et les contextes ont été révélés par une inspection des interfaces des outils de simulations de l'entreprise et une lecture de l'outil de suivi de bogues et de gestion des incidents utilisé chez CAE, Jira : tous les problèmes relevés par les membres de l'entreprise y sont décrits en profondeur et classés par fréquence et criticité. C'est un excellent aperçu de la « santé » des logiciels.

Les solutions à ces problèmes sont la synthèse de nombreux éléments :

- La revue de littérature présentée plus tôt, pour s'inspirer des patrons des bibliothèques existantes et voir ce que d'autres experts en ingénierie des facteurs humains pensent être important, aidant ainsi à définir une granularité pour nos patrons,

- Un examen de l'existant, combinant une revue de littérature hors patrons avec une veille technologique,
- Et bien sûr les bonnes pratiques du personnel de l'entreprise.

2.4.2 Rédaction d'un matériel adapté aux développeurs logiciels

Le matériel précédemment relevé a ensuite été rédigé en patrons proprement dits selon les conseils de Meszaros (1997) et en prenant soin d'ajuster l'écriture au besoin et à la culture de l'organisation.

La *rédaction adaptée aux développeurs logiciels* comporte trois facettes : le choix du vocabulaire (et le lexique associé), un style d'écriture prescriptif et l'importance donnée aux exemples. Ces trois mesures sont destinées à rendre la bibliothèque de patrons efficace et agréable à un lectorat néophyte en ergonomie.

Le *choix du vocabulaire* consiste à repérer les termes ou notions qui posent problème aux développeurs et à les remplacer par un vocable adapté. C'est par exemple pour cette raison que dans la description d'un patron l'abscons *Forces en conflits* a été écarté au profit d'un libellé *Contexte*. Le nom de nos modes de présentation a été affiné au fil des retours utilisateurs : les *archétypes* sont devenus des *idiomes*, puis des *écrans standards* et enfin des *types d'applications*, plus explicites pour les développeurs. De même le *nuancier* a été rebaptisé *couleurs* et les *miniatures de patrons* sont devenues des *aperçus de patrons* pour que les utilisateurs ne confondent plus l'extrait de patron avec son illustration. Cela nous permet également un léger glissement sémantique : une *miniature* indiquait une version complète mais réduite, alors qu'*aperçu* suggère une version partielle d'une entité plus riche, comme c'est effectivement le cas. Choisir et contrôler le vocabulaire nécessitent de considérer un lexique comme une pierre angulaire de la bibliothèque et non comme une fonction auxiliaire et contraignante. Un responsable de CAE commentait ainsi le lexique lors d'une séance de test : « C'est notre plus grand défi. (...) Les gens ont des choses différentes dans la tête pour le même mot, les ontologies c'est pas évident ».

Le *style d'écriture prescriptif* est très semblable à la rédaction de procédures, on se référera avec profit à la littérature idoine (Wieringa, 1999). Celle-ci recommande d'écrire des phrases courtes, impératives, au style direct, avec des énoncés affirmatifs utilisant le nom des menus ou des

commandes tels que l'utilisateur les voit à l'écran. Nous y ajouterons quelques éléments de rédaction de manuels, comme le fait d'écrire sous forme de listes à points, de mettre les termes importants en gras et d'inclure toute l'information là où elle est requise en la répétant au besoin. Des séances d'essais informels ont de plus révélé la nécessité de donner un titre à chaque section ou sous-section du patron. Dans le cas contraire, les développeurs sont dans l'incapacité de balayer rapidement la page et, ne sachant pas ce qu'ils peuvent y trouver, la quittent rapidement.

Enfin, *l'importance des exemples* se traduit par trois qualités. Premièrement, la grande quantité d'exemples accompagnant chaque patron, environ une dizaine, essentiellement sous forme de captures d'écran. Deuxièmement, le soin apporté au choix de ces exemples : pertinence de l'image (illustration directe du patron, sans ambiguïté, distraction ni mauvaise pratique latente), qualité (résolution, esthétique), variété (diversité des plates-formes technologiques et des contextes d'usage, avec autant que possible un exemple local à CAE). Troisièmement par le traitement qui en est fait : recadrage, composition de plusieurs captures pour illustrer une interaction ou mise en valeur de certaines zones.

2.4.3 Développement du système de gestion de contenu

En parallèle du recensement et de la rédaction des patrons, nous avons donc entrepris de mettre en place un système de gestion de contenu spécifique aux patrons de conception, à l'usage de CAE. Après avoir essayé de nombreux systèmes conceptuellement et fonctionnellement différents (annexe B.1), nous avons finalement opté pour un moteur de blogue en source libre, modifié pour mieux répondre à nos besoins, Wordpress.

Nous l'avons peu modifié pour des raisons de maintenabilité, à l'exception notable de l'ajout d'un module Markdown permettant d'écrire les patrons dans un format de texte simple, facile à lire et à écrire, puis les convertissant en HTML valide (Gruber, 2004).

Une description complète de l'implémentation technique du système de gestion de contenu est disponible dans l'annexe C.

2.4.4 Itération et finalisation

Le processus de recensement, rédaction et diffusion des patrons a été vérifié et validé à intervalles réguliers par les quatre membres de l'équipe d'ingénierie des facteurs humains et par

la dizaine de membres du comité des architectes de l'entreprise, responsable et garant des choix technologiques et des processus mis en œuvre chez CAE. Les patrons eux-mêmes ont été activement lus, commentés et appliqués par plus de 50 ingénieurs ou spécialistes des facteurs humains, dont le feedback nous a permis de faire des itérations informées.

Nous avons également profité de leurs commentaires pour définir la spécificité de nos patrons, requis pour estimer ultérieurement leur pertinence pour une interface donnée (l'autre aspect étant l'applicabilité, comme expliqué dans la section 1.3.1 *Pas de guidage dans les bibliothèques de patrons*). Nous considérons nos patrons comme étant d'égale qualité, mais de spécificité variable : certains patrons sont suffisamment génériques pour marcher dans plusieurs circonstances, alors que d'autres ne fonctionnent que dans des cas de figure très particuliers. La spécificité de chaque patron a été établie et validée conjointement par les membres de l'équipe d'ingénierie des facteurs humains selon l'échelle présentée dans le tableau 2-1, les notes finalement attribuées aux patrons sont présentées dans l'annexe C.1. La validation des notes par plusieurs personnes fait que la valeur de spécificité d'un patron, et par conséquent de pertinence, est relativement fiable.

Tableau 2-1 : Échelle de spécificité des patrons de conception CAE

Note	Critère
1	Fonctionne dans la plupart des cas
2	Modérément commun
3	Uniquement dans des cas particuliers

La spécificité est une dimension importante car c'est une bonne indication de la valeur qu'ajoute ou non un patron à une conception : des recommandations sur les tailles ou les couleurs sont utiles mais superficielles comparativement à des solutions d'architecture de l'information ou de soutien du flux de l'utilisateur, à même de débloquer un problème de conception ou de fournir

une solution holistique. Par exemple, le *Dévoilement progressif*⁸ va profiter à l'utilisateur à travers toute l'application et avoir des répercussions au-delà d'un écran ou d'un panneau unique.

À l'issue de plusieurs cycles de corrections et d'assurance qualité, nous disposons d'une trentaine de patrons (annexe C.1) capturant les bonnes pratiques de l'entreprise et d'un outil informatique pour les diffuser sur son réseau intranet. La diffusion effective des patrons est cependant suspendue, et seuls les professionnels susmentionnés y ont accès dans un premier temps : nous pourrions de la sorte évaluer ultérieurement l'expérience « premier contact », sans apprentissage, des développeurs logiciels avec la bibliothèque.

Une limitation notable de nos patrons à ce stade est qu'ils forment une bibliothèque mais pas un langage de patrons : il leur manque pour cela un système de valeur fort et un principe structurant en complément de leurs relations horizontales.

2.5 Développement des nouveaux modes de présentation

Nous venons de récapituler la rédaction des patrons proprement dits de la bibliothèque de CAE, mais la majeure partie des efforts relatés dans ce mémoire s'applique à améliorer la navigation et la découverte dans une bibliothèque de patrons. Ce chapitre présente donc les trois modes de présentation d'information mis au point pour rendre la bibliothèque des patrons de conception accessible aux développeurs logiciels : *arbres de décision*, *types d'application* et *aperçus de patrons*.

Le raisonnement derrière le choix et l'exécution de chaque méthode sont explicités afin que le lecteur puisse juger de leur pertinence selon son besoin. Une première séance de validation semi-formelle menée à l'issue de la mise au point de ces modes et les actions correctives qui ont suivi sont également évoquées.

2.5.1 Types d'applications

Les *types d'applications* sont des représentations visuelles et textuelles de stéréotypes de grandes familles de logiciels, à mi-chemin entre les *postures* définies par Welie & van der Veer (2003), et

⁸ *Progressive disclosure*, tiré de la bibliothèque Designing Interfaces (Tidwell, 2005).

les *Standard Screen Patterns* définis par Neil (2010). Welie décrit les postures (figure 2-1) de la manière suivante :

« D'expérience nous savons que beaucoup de sites sont en fait assez similaires en ce qu'ils partagent les mêmes buts (...) Ceci peut être appelé la +posture+ du site. Par exemple, les petits sites corporatifs sont souvent tellement similaires qu'on peut écrire des patrons les décrivant. La même chose marche pour les sites de d'information, les sites communautaires et ainsi de suite. On peut définir plusieurs de ces postures de sites qui peuvent être prises comme base pour la conception de nouveaux sites » (traduction libre de l'auteur).

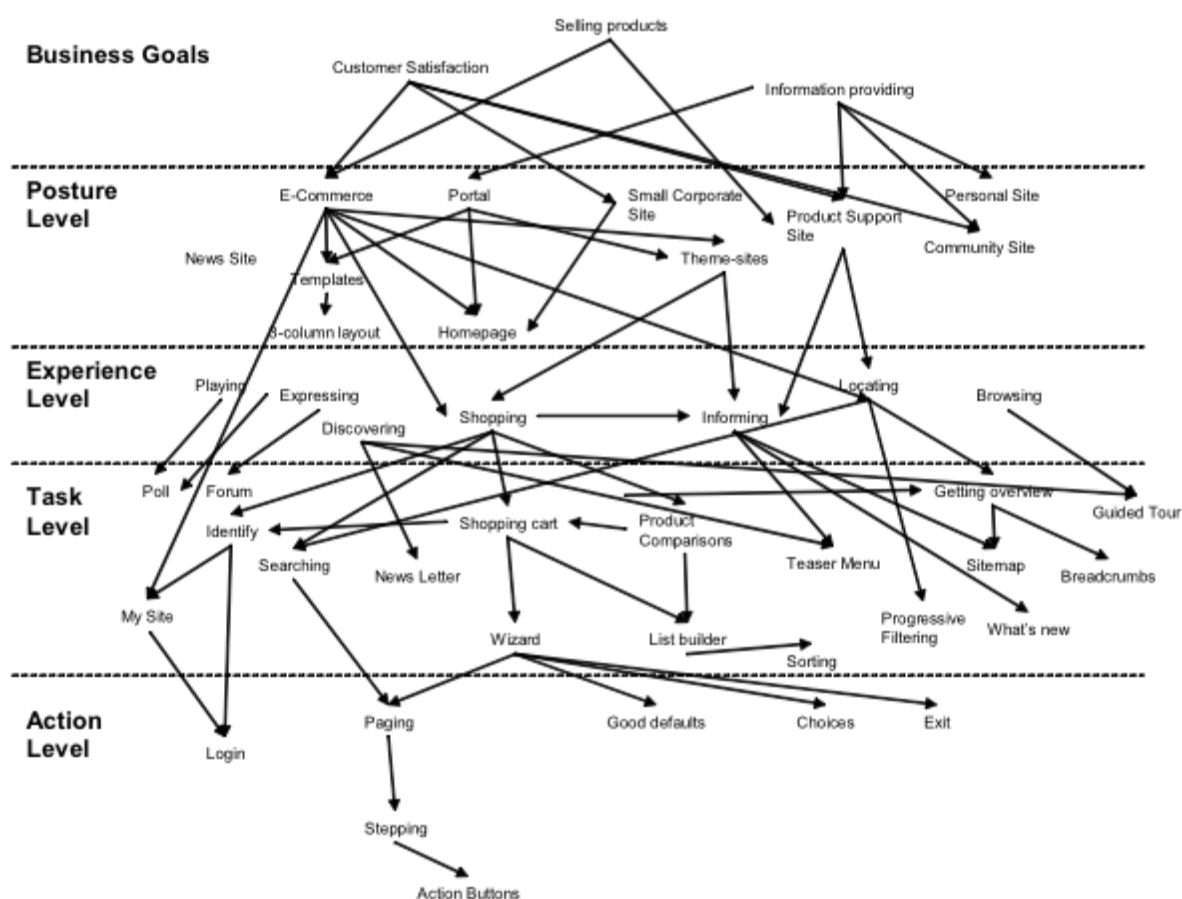


Figure 2-1: Postures pour sites Web de magasinage en ligne

Les Standard Screen Patterns (figure 2-2) sont tout simplement un recensement des types de logiciels les plus communs :

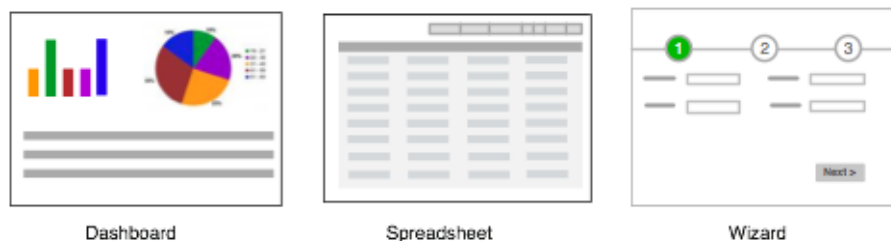


Figure 2-2: Standard Screen Patterns (extraits)

Les types d'applications sont des archétypes génériques de logiciels fréquents auxquels on associe une collection prédéfinie de patrons. Ce sont des artefacts assez faciles à appréhender pour l'utilisateur car ils suivent un modèle mental lié au grand public, et renforcé par la terminologie des gammes commerciales des éditeurs logiciels, plutôt qu'au monde de l'ergonomie. On identifie ainsi aisément navigateurs Web, chiffriers, lecteurs multimédia ou clients de messagerie instantanée comme autant de types d'applications. Cependant, la spécificité des activités de CAE ne nous permet pas de recourir à cette nomenclature grand public ; aussi avons-nous compulsé la documentation des produits de l'entreprise pour repérer les écrans fonctionnellement et visuellement similaires. Des types d'applications pertinents pour les logiciels CAE ont été identifiés (annexe C.2), parmi lesquels se trouvent les systèmes de gestion de contenu, des cockpits virtuels de toutes sortes (figure 2-3), formulaires, etc.

En plus de la simplicité conceptuelle évoquée ci-dessus, la quantité de types d'applications est inférieure à la quantité de patrons (dans un rapport estimé de un à 10 chez CAE), ce qui leur permet d'agir comme un raccourci pour les développeurs dans les tâches de recherche et d'identification de patrons.

Les types d'applications ressemblent un peu aux patrons de conception en ce qu'ils font appel au même principe de pairage de patrons et qu'ils pointent vers d'autres patrons de conception en justifiant leur choix. Ils en diffèrent cependant en ce qu'ils capturent ce qui est existant dans l'entreprise et pas uniquement ce qui est bon.

Les types d'applications prennent la forme de texte illustré, disponible sur l'intranet de CAE à l'instar des arbres de décision. Ces deux modes de présentation ne sont cependant pas parfaitement substituables l'un à l'autre. Ils ne visent pas forcément les mêmes sous-ensembles de patrons et ne répondent pas aux mêmes besoins des usagers, comme l'a si bien exprimé un de nos participants lors des essais semi-formels⁹ : « Ça dépend de où tu en es rendu. Avec les arbres on peut *fine-tuner*, avec les types d'application, c'est plutôt *from scratch* ».

⁹ Nous retrouverons cette attitude pendant les évaluations formelles. La plupart des participants y ont exprimé leur appréciation de plusieurs modes en même temps et nous ont enjoint de ne pas en choisir un au détriment des autres.

Modèle immersif



Google Maps, Half-Life, CATIA, SolidWorks...

Toutes les cartes des IOS, VSIM, le réglage des caméras par rapport aux avions, l'orientation d'une tourelle de char.

L'utilisateur veut agir sur une représentation directe du monde réel (une carte, un avion) par opposition à travailler sur des données purement "informatiques" (des fichiers à copier par exemple). L'objet représenté est :

- Simple, ou du moins bien compris par l'utilisateur,
- Physiquement palpable (pas une tempête ou une condition météo par exemple).

L'utilisateur veut effectuer une configuration, une correction ou accomplir des opérations dans l'espace. Par exemple : le repérage d'une entité sur une carte ou son placement par rapport à une autre entité.

La nature ludique des modèles interactifs favorise les [Opérations dans le contenu](#) en général et les [Manipulations directes](#) en particulier. Pensez à mettre un maximum de [Commandes contextuelles](#) et dissimuler les propriétés des entités dans des [Backstages](#).

Il est également assez facile d'y mettre des [Zooms sémantiques](#) et autres [Transitions plein écran](#) pour gérer la granularité des commandes et des informations.

L'immersion est à double tranchant : l'interface s'éloigne des conventions informatiques à mesure qu'elle s'approche de la réalité. Pour éviter que ce ne soit contre-productif, ne pas oublier les [Issues de secours](#) et [Le complexe possible](#).

Figure 2-3: Type d'applications *Modèle Immersif*

2.5.2 Arbres de décision

Les *arbres de décision* tentent de répondre à un problème précis qu’aurait le développeur logiciel avec son produit. Pour ce faire, le développeur répond à une succession de questions sur les caractéristiques ou symptômes qu’il observe dans son interface, jusqu’à ce qu’il arrive à un patron. Il lui est plus facile de répondre « passivement » à des questions formulées à l’avance que de transcrire « activement » des symptômes en besoins utilisateurs puis en solutions adaptées. Le développeur évite ainsi au maximum les activités de recherche, lecture ou comparaison de patrons, dont les lourdes exigences en temps et en expertise ont été montrées dans le chapitre 1.3 *Les difficultés des développeurs logiciels*.

Les arbres de décision prennent deux formes dans la bibliothèque de CAE. Ils sont d’abord présentés comme des assistants étape-par-étape codés en JavaScript (figure 2-4). Un panneau présente la première question et les réponses possibles. Quand l’utilisateur clique sur la réponse souhaitée, un nouveau panneau remplace le précédent avec de nouvelles questions et réponses, dépendant de la réponse précédemment choisie, et ainsi de suite jusqu’à ce qu’à ce que l’utilisateur ait répondu à assez de questions pour que la bibliothèque puisse lui proposer un patron pertinent. Les arbres sont également présentés sous forme de diagramme UML (figure 2-5), faisant appel exactement aux mêmes questions et réponses que précédemment mais les affichant sous une forme plus graphique que l’utilisateur peut parcourir à son gré. Cette forme a l’avantage d’exposer une partie des relations entre patrons : un nœud sur le diagramme correspond au choix exclusif d’un patron plutôt que d’un autre, établissant une relation de compétition entre ces deux patrons.

La mise en place des arbres de décision correspond à un besoin précis au sein de l’entreprise : la résolution des problèmes d’interface utilisateur rencontrés par les clients de CAE et révélés par les outils de suivi de bogues ou par l’assurance qualité de l’entreprise. Les arbres de décision répondent à la question ainsi formulée par un développeur logiciel de CAE : « Comment [l’équipe d’ingénierie des facteurs humains] peut m’aider pour mes *snags* JIRA ? »

Dans les faits, la rédaction des arbres a débuté par l'identification des problèmes récurrents dans les interfaces CAE, en assistant aux réunions de développement ou de suivi de produit, compulsant l'outil de suivi de bogues et effectuant des inspections heuristiques. Tous les problèmes dégagés n'ont pas pu donner lieu à des arbres car nous n'avions pas toujours assez de patrons sur un problème précis pour le couvrir de manière satisfaisante pour l'utilisateur. Ce revers révèle en revanche une autre utilité des arbres de décision : la mise en évidence des lacunes dans la couverture de la bibliothèque de patrons.

Figure 2-4: Arbre de décision *Limiter le taux d'erreur* (assistant, extraits)

De haut en bas et de droite à gauche : panneau présenté par défaut, puis panneau présenté si l'utilisateur choisi « Oui » à la première question, « Non » à la deuxième question et « Non » à la troisième question.

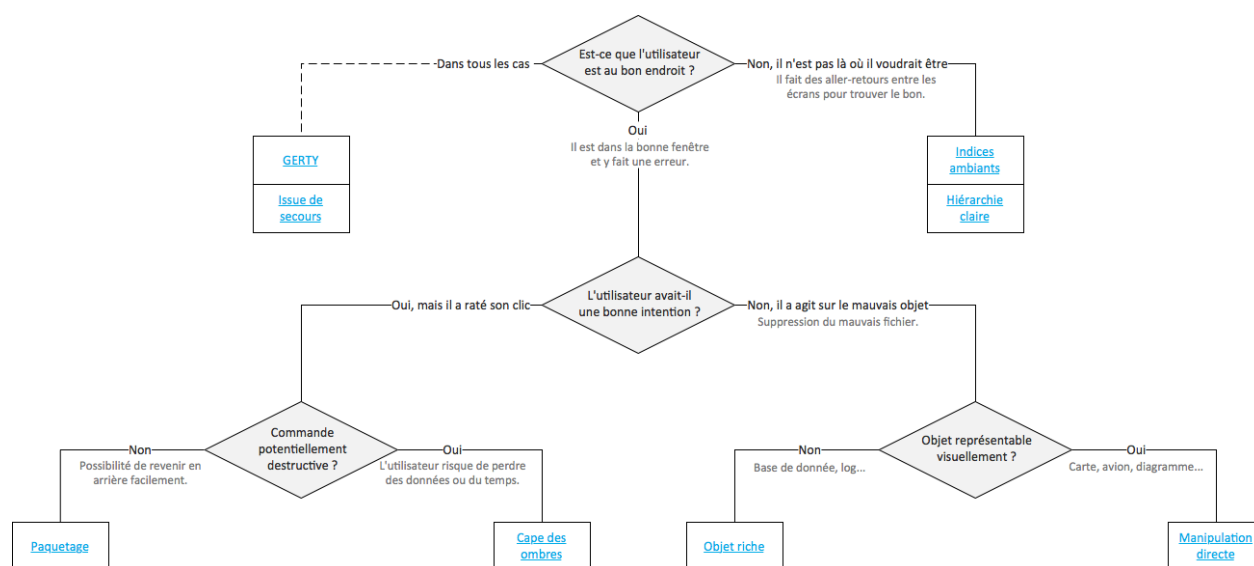


Figure 2-5: Arbre de décision *Limiter le taux d'erreur* (diagramme)

Nous concluons cette section en insistant sur la nature systémique des arbres de décision, l'utilisateur y étant guidé étroitement. C'est le mode de navigation qui donne le plus de levier à l'équipe d'ingénierie des facteurs humains, qui est en mesure de vraiment polir l'aide qu'elle fournit par ce biais. La liste des arbres de décision est disponible dans l'annexe C.3.

2.5.3 Aperçus de patrons

Les *aperçus de patrons* sont basés sur l'idée qu'on peut évoquer un patron par un fragment d'information riche (Meszaros & Doble, 1997). Ils sont composés de trois éléments : le titre du patron, un visuel schématisé de la solution et une description de la solution en quelques mots (figure 2-6).

Les aperçus ont été conçus pour répondre à plusieurs exigences formulées empiriquement suite à notre utilisation personnelle des bibliothèques existantes :

- Fournir les éléments nécessaires à la compréhension d'un patron directement dans ou à proximité de celui-ci afin de ne pas interrompre la lecture. En effet comme un patron fait référence à d'autres patrons dans son corps de texte, il contraint l'utilisateur à aller et

venir en permanence dans la bibliothèque pour lire ces patrons auxiliaires et être en mesure d'exploiter pleinement le patron d'origine.

- Fournir un raccourci cognitif à la compréhension, au moins partielle, du patron,
- Engager l'utilisateur en lui présentant d'office la réponse à *Qu'est ce qu'il y a pour moi là-dedans ?*¹⁰,
- Répondre aux exigences précédentes dans un format suffisamment compact (physiquement et cognitivement) pour pouvoir être affiché ponctuellement dans les autres modes de navigation. Avoir un artefact concret et compact permet aussi à l'utilisateur de stocker plus facilement plusieurs aperçus dans sa mémoire de travail, une capacité très utile lorsque l'utilisateur doit agréger plusieurs patrons pour en comprendre un seul comme expliqué précédemment.
- Répondre aux exigences précédentes par un format suffisamment riche pour qu'il puisse être traité en *reconnaissance plutôt que rappel* (Nielsen, 2005) par l'utilisateur, contrairement aux libellés de texte simple utilisés d'ordinaire.

L'aperçu est illustré avec un visuel schématisé. Nous craignons qu'une capture d'écran d'un système spécifique dissuade les développeurs de lire le patron, le croyant réservé à une technologie en particulier. De même, un visuel vague évite de recommander une granularité ou une implémentation en particulier au détriment des autres et aide à communiquer la nature « abstraite mais instanciable » du patron. L'usage de captures d'écran pourrait cependant s'avérer pertinent dans le cadre d'une organisation travaillant avec moins de plates-formes et de technologies que CAE. Les visuels en question ont été dessinés à l'aide d'un logiciel de dessin vectoriel et corrigés plusieurs fois d'après les commentaires des utilisateurs.

Les aperçus sont pour le moment affichés tous ensemble dans une page d'archive dédiée faisant office d'index du contenu de la bibliothèque, ou individuellement comme en-tête d'un patron lorsqu'on en consulte la page. Nous espérons à terme pouvoir les afficher par des vignettes ou en

¹⁰ C'est une forme d'engagement progressif. On présente d'office plus que ce que l'utilisateur a demandé pour lui mettre « le doigt dans l'engrenage », mais suffisamment peu pour qu'il le lise quand même en entier par défaut.

note de bas de page dès qu'un patron est mentionné dans la bibliothèque, à quelque endroit que ce soit.

En outre, le texte d'un aperçu reprend des éléments de la *solution* plutôt que du *problème* ou du *contexte* d'un patron. C'est une qualité souhaitée : les aperçus forment ainsi les *réponses* naturelles, et étapes terminales, aux questions que sont les arbres de décision et les types d'applications, lesquels spécifient plutôt des *problèmes* ou des *contextes*.

Transition plein-écran

By JOCELYN | Published: NOVEMBER 25, 2010 | Edit

Mettre une fonction ou un mode dans un écran à part mais intégré dans le contenu du logiciel.

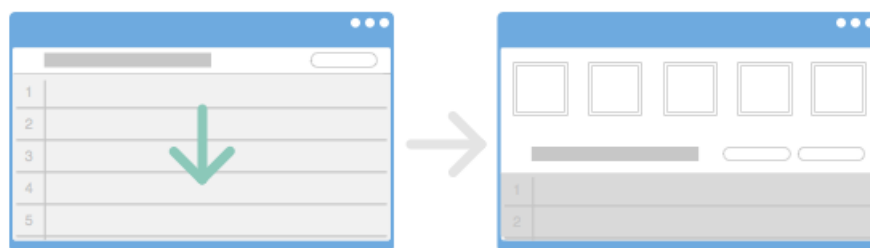


Figure 2-6: Aperçu du patron *Transition plein-écran*

2.5.4 Vérification et validation

Ces trois modes de présentation de l'information ont été validés par le feedback constant des professionnels de CAE. Nous avons en outre conduits quelques brèves séances d'essais préliminaires dans deux buts :

1. Vérifier la valeur de nos propositions,
2. Rôder le protocole expérimental.

Une dizaine de développeurs logiciels, sans formation en ergonomie ni connaissance préalable de la bibliothèque, ont donc dû chercher des patrons de conception spécifiques dans la bibliothèque de CAE à l'aide des trois nouveaux modes de présentation, leurs temps étant chronométrés et

leurs impressions recueillies pendant et après la séance. Leur participation nous a permis d'élaborer une méthodologie pertinente telle que décrite en détail dans le chapitre suivant. Nous avons également utilisé ces contributions pour dégrossir nos propositions :

- De nombreux défauts dans la définition originelle des arbres de décision ont été révélés. Ceux-ci n'étaient d'abord disponibles que sous forme de diagramme ; nous pensions que cette forme serait familière aux développeurs logiciels en raison de sa similarité avec les diagrammes UML enseignés en génie logiciel et génie informatique. En pratique les utilisateurs étaient surpris de trouver ce type de contenu sur un site Web et se sentaient submergés par la complexité de la chose. De même, l'affordance des liens menant vers les patrons depuis les arbres était faible ; les utilisateurs n'ont quasiment jamais vu qu'ils pouvaient aller sur le patron qui leur était proposé à la fin de l'arbre. Nous avons donc corrigé et mis en retrait les diagrammes UML, et fourni à la place les assistants présentés précédemment.
- Le style de rédaction et le vocabulaire employés dans les types d'application ont été ajustés,
- Les illustrations des aperçus et les titres des patrons ont été corrigés plusieurs fois.

Il convient de préciser que les nouveaux modes de présentation nécessitent un surcroît de rédaction, de programmation et de dessin. L'adéquation de la rédaction à l'audience visée, la recherche et le traitement des exemples sont également des tâches chronophages. Toutes choses étant égales par ailleurs, notamment la finitude des ressources en personnel, temps ou budget, cette charge de travail supplémentaire peut contraindre l'équipe responsable de la bibliothèque à écrire moins de patrons *per se* afin de rendre le corpus existant accessible aux utilisateurs de la bibliothèque. En outre, écrire le contenu d'un arbre de décision ou d'un type d'applications nécessite approximativement le même effort de la part du rédacteur. Les arbres sont toutefois plus longs à parachever puisqu'ils nécessitent en sus des tâches de dessin et de programmation pour transcrire le contenu original sous forme graphique et sous forme interactive. Comme en plus ils s'appliquent mieux à des problèmes précis, il est intéressant de les livrer après les autres modes, une fois que la bibliothèque est stable et que les besoins et les processus de l'organisation sont bien compris, afin de ne pas gaspiller des ressources dans des tâches de mise en forme.

Signalons enfin que nos patrons ne sont pas cloisonnés entre différents modes de présentation, la plupart sont mêmes référencés dans tous les modes. En conséquence, il n'y a pas de variation dans la rédaction des patrons en fonction de leur apparition dans un mode de présentation ou un autre : tous nos patrons sont écrits avec le même style et la même substance. Les modes de présentation viennent se brancher ultérieurement sur ce corpus indivisible.

À ce stade, nous disposons de plusieurs occurrences des trois nouveaux modes de présentation, et les avons appliquées à nos patrons existants. Nous allons maintenant évaluer l'influence de ces modes de présentation sur la capacité des développeurs logiciels à trouver de bons patrons d'IHM.

CHAPITRE 3 ÉVALUATION EXPÉRIMENTALE DES NOUVEAUX MODES DE PRÉSENTATION

Ce chapitre décrit la nature et le détail de l'expérimentation qui a été menée pour évaluer l'effet des propositions décrites précédemment. Nous commençons par exposer la méthodologie employée, suivie des résultats. Ces résultats sont ensuite analysés et des recommandations sont proposées pour application immédiate. Un recueil complet des résultats d'entrevue est disponible dans l'annexe D.

Les expérimentations ont donc consisté en l'exécution par les participants d'une même tâche de recherche de patron avec quatre modes de présentation, l'un d'entre eux (index alphabétique des patrons) faisant office de référence.

Précisons que ces expérimentations ont volontairement évalué le tout premier contact des participants avec les patrons de conception d'interface utilisateur en général et la bibliothèque de patrons de CAE en particulier : nos expériences portent donc sur des utilisateurs novices. Cette contrainte a été appliquée pour des raisons organisationnelles : en raison du surnombre précédemment évoqué de développeurs logiciels par rapport aux ergonomes chez CAE, il est raisonnable de penser qu'un développeur puisse se servir de la bibliothèque sans avoir reçu de formation préalable. Même si dans les faits l'ouverture publique de la bibliothèque de patrons de CAE devrait être accompagnée d'annonces et de publications sur l'intranet, d'ateliers de sensibilisation et de formation, il a semblé préférable de majorer le risque afin de parer le pire des cas.

Il aurait été possible de procéder autrement dans une autre organisation avec un effectif total plus faible, moins dispersé géographiquement ou moins déséquilibré dans sa répartition entre développeurs et ergonomes. Il est probable qu'alors toutes les parties prenantes du processus de développement logiciel (développeurs, gestionnaires, responsables commerciaux) auraient été conviées à des séances de présentation et d'entraînement à l'usage de la bibliothèque de patrons. Les mêmes modes de présentation auraient été testés, de la même manière, mais avec des utilisateurs experts bénéficiant d'un effet d'apprentissage.

3.1 Méthodologie

3.1.1 Participants

Dix employés de CAE, âgés entre 28 et 40 ans, ont participé à l'étude. Nous les présenterons en trois catégories :

- Quatre d'entre eux sont des développeurs logiciels,
- Quatre ont des responsabilités de gestion d'équipe ou de projet¹¹,
- Deux effectuent respectivement des tâches d'assurance qualité et de design graphique.

Leur inaptitude à concevoir des interfaces utilisateur est notre *variable contrôlée*. Nous nous sommes assurés qu'aucun des participants n'avait de formation ou de compétence en conception d'interface humain-ordinateur, n'avait entendu parler de patron de conception d'interfaces utilisateur, ni ne s'était servi de la bibliothèque de patrons CAE au préalable.

Les participants ont été directement sollicités par l'auteur et s'engageaient sur la base du volontariat, sans compensation pour leur participation à l'étude. Ils ont été recrutés dans différentes divisions de l'entreprise (civil, militaire, aviation et santé) pour éviter de biaiser les résultats. Aucun ergonome n'a participé aux tests, de manière à ne pas en fausser les résultats à la hausse.

3.1.2 Tâches

Les participants devaient sélectionner un écran d'un produit CAE de leur choix, afin d'être en terrain familier, limitant l'influence de ce facteur exogène. Ils devaient trouver, pour cet écran, un patron *approprié* avec chacun des quatre modes de présentation (nos *variables indépendantes*). Un patron était considéré *approprié* s'il améliorait significativement l'écran choisi ou s'il était déjà présent et appliqué correctement dans cette interface. Les quatre modes de présentation faisaient appel à la même bibliothèque en arrière-plan.

¹¹ Une multitude de postes tels que chef d'équipe, chef de produit ou propriétaire de produit correspondent peu ou prou au même profil de « gestion de projet ».

Il était demandé de choisir un écran seul et non un produit au complet pour restreindre le nombre de patrons appropriés et ainsi éviter les faux positifs. Inversement, l'écran initialement choisi par le participant pouvait être rejeté par l'examineur si aucun patron existant ne pouvait s'y appliquer, l'évaluation portant sur l'efficacité des modes de présentation de la bibliothèque, pas son ampleur. Pour des raisons de sécurité dans l'entreprise, les participants n'avaient souvent accès qu'aux logiciels sur lesquels ils étaient en train de travailler ; leur choix s'est donc spontanément porté sur des écrans de ces produits. En conséquence, les participants connaissaient très bien les écrans qu'ils choisissaient.

On ne demandait pas aux participants d'instancier effectivement le patron dans l'interface choisie : d'une part parce que l'étude porte sur la navigation vers un patron plutôt que l'implémentation dudit patron, d'autre part parce que les développeurs réfléchissent toujours, par réflexe professionnel, à l'implémentation technique du patron et auraient pu favoriser des patrons plus faciles à programmer mais moins pertinents vis-à-vis des questions d'utilisabilité.

3.1.3 Écrans choisis par les participants

Les participants ont tous effectué l'expérience sur des projets en cours de développement et sur lesquels ils travaillaient au moment des essais. La politique de sécurité et de confidentialité de l'entreprise interdisait le choix d'autres projets.

Les écrans effectivement choisis dans chacun de ces projets sont décrits dans l'annexe D.5.

3.1.4 Patrons de conception

La bibliothèque contenait 27 patrons au moment des tests (annexe C.1). Un patron trouvé avec un des trois modes de présentation ne pouvait pas être utilisé avec un autre mode de présentation, pour éviter que les participants ne biaisent l'expérience en se jetant sur un choix sécuritaire. En échange, l'examineur vérifiait au préalable que plusieurs patrons soient applicables à l'écran choisi, de sorte que le participant ait la possibilité d'aboutir à quatre patrons différents.

Les modes de présentation étaient présentés et ont été choisis par les participants sans ordre particulier, avec cependant une tendance en faveur de l'agencement « physique » des modes sur la page : les arbres de décision, les types d'applications, l'index des aperçus et l'index

alphabétique (mode de référence) ont été choisis respectivement en première, deuxième, troisième et quatrième position près de 50% du temps (annexe D.6).

3.1.5 Système informatique utilisé

Les participants utilisaient leurs ordinateurs de production (fournis à titre individuel par CAE) pour les tests. Leurs configurations matérielles sont comparables : ordinateurs portables, reliés ou non à un moniteur externe, ou ordinateurs de bureau reliés à deux moniteurs externes. Dans tous les cas, les participants ne dédiaient au site intranet de la bibliothèque qu'un moniteur, sa diagonale variant de 13 à 19 pouces, pour des résolutions respectives de 1024×768 pixels à 1440×900 pixels.

Tous les ordinateurs exécutaient Windows XP. Plusieurs fureteurs ont été utilisés, sans différence notable sur l'apparence ou le comportement de la bibliothèque : Chrome 7, Firefox 3, Internet Explorer 8 et Opéra 9.

3.1.6 Données recueillies

Les données suivantes ont été recueillies pour chaque mode de présentation :

- La réussite dans la recherche d'un patron,
- Le temps requis pour trouver un patron approprié, d'un mode relativement à l'autre,
- La satisfaction de l'utilisateur à propos du mode de présentation, évaluée sur une échelle de Likert en 5 points à propos du mode (annexe E),
- La pertinence du patron trouvé (évaluée par l'examineur : spécificité, définie au préalable, et applicabilité) par rapport à l'interface CAE initialement choisie.

Ces quatre premiers éléments sont nos *variables dépendantes*. La satisfaction des utilisateurs et l'applicabilité du patron trouvé ont été recueillies lors d'une entrevue semi-structurée à la fin du test. Il était en outre demandé aux participants de penser leurs actions à voix haute pendant la séance de test. Leur verbalisation a été prise en note manuellement.

Il est arrivé plusieurs fois que le participant discute ou réfléchisse sur l'utilisabilité de l'écran sélectionné à la lumière des informations acquises en lisant la bibliothèque, pendant ou après la sélection d'un patron. Le chronomètre a été arrêté lors de ces occurrences.

3.1.7 Procédure

Les participants sont passés un par un. Au commencement de chaque session de test, le participant était accueilli, l'objectif de l'étude lui était expliqué de vive voix et un questionnaire biographique succinct était rempli (voir annexe D.2). Il était informé des garanties d'anonymat et d'impunité. De brèves instructions sur les tâches à effectuer et l'enchaînement des événements durant la session lui étaient ensuite communiquées.

Le participant devait répondre à un questionnaire post-expérimentation et une entrevue semi-structurée conduite par l'expérimentateur après qu'il eût réalisé ces tâches.

Les expérimentations se sont déroulées dans les locaux de CAE, au bureau de chaque participant, en présence de l'expérimentateur qui observait le travail fait par le participant, enregistrait sa verbalisation, collectait le questionnaire et posait les questions.

3.2 Résultats

Les résultats portent sur le taux de succès dans la recherche de patrons, et le temps de découverte d'un patron, la satisfaction des utilisateurs et la pertinence des patrons trouvés par rapport aux écrans choisis. En second lieu, une corrélation est faite entre ces observations et les réponses obtenues via le questionnaire biographique.

Notons que notre expérimentation a porté sur une bibliothèque de 30 patrons, or certaines bibliothèques en contiennent jusqu'à 125. Notre validation expérimentale, notamment la performance relative des différents modes de présentation, aurait intérêt à être vérifiée par une étude de plus grande ampleur.

3.2.1 Taux de succès dans la recherche de patrons

Les participants ont trouvé des patrons qu'ils jugeaient appropriés 37 fois sur 40 tentatives, les trois abandons étant directement imputables à des interruptions de la session de test pour motif professionnel (irruption d'un collègue par exemple).

3.2.2 Temps de recherche

Le temps de recherche d'un patron est plus élevé avec les nouveaux modes de présentation (figure 3-1). Il augmente de 51% avec les types d'applications (de 3:48 min à 4:41 min), de 23%

avec les arbres de décision (de 3:06 min à 3:26 min) et de 11% avec les aperçus (de 3:06 min à 3:26 min).

Ce chiffre cache cependant des comportements différents. Si les participants ont utilisé leur temps avec l'index alphabétique pour effectuer des recherches aveugles, aléatoires, de type essai-et-erreur, ils ont par contre utilisé leur temps avec les nouveaux modes pour lire des détails supplémentaires ou réfléchir à voix haute au problème auquel ils étaient confrontés et trouver la meilleure manière d'y répondre.

Notez que ces temps sont indicatifs de la performance relative des modes de présentation les uns par rapport aux autres, et non de leur performance absolue puisqu'il était demandé aux participants de verbaliser leurs actions, handicapant leur vitesse de travail. La régularité de l'écart-type d'un mode de présentation à l'autre ($\pm 10\%$) semble indiquer que le handicap de la verbalisation est relativement constant pour tous les modes de présentation ; leur comparaison semble donc valide.

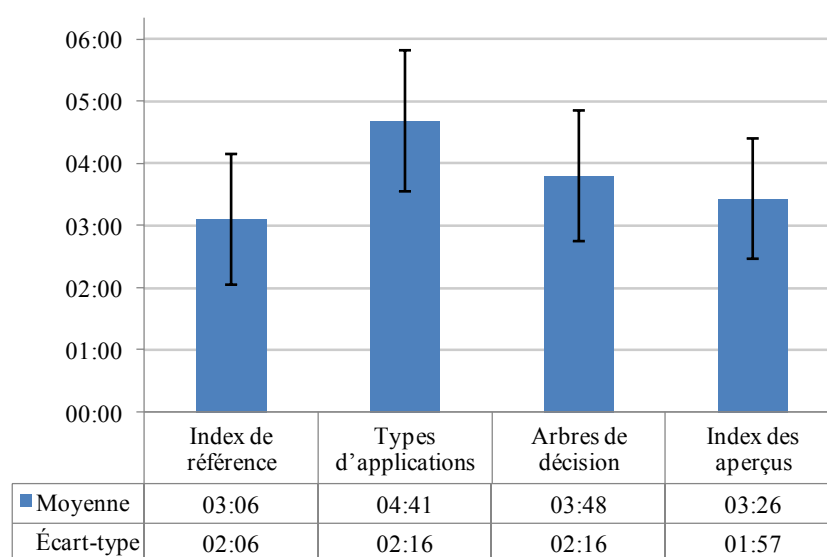


Figure 3-1: Temps requis (en minutes) pour trouver un patron

L'écart-type est très stable, à environ une minute pour chaque mode. Cela s'explique par l'homogénéité du comportement des participants lors de la navigation dans la bibliothèque de patrons : une minorité (environ un tiers) des participants se sont jetés sur le premier patron fourni

par le mode de navigation. En revanche la majorité des participants ont lu entre deux et trois patrons¹² en mode essai-et-erreur avant d'en déclarer un approprié.

La constance de la dispersion pour les quatre modes de présentation nous permet d'affirmer que les utilisateurs d'une bibliothèque de patrons tolèrent au maximum trois tentatives avant d'abandonner, c'est-à-dire prendre le patron en cours quel que soit son degré d'adéquation à la situation.

Par ailleurs, nous n'avons pas constaté d'effet d'apprentissage significatif lors des essais. Au mieux, les participants reconnaissaient le nom de patrons déjà consultés mais devaient quand même les relire pour se rappeler leur contenu.

3.2.3 Satisfaction des utilisateurs

La satisfaction des utilisateurs est plus élevée avec les trois nouveaux modes de présentation qu'avec l'index alphabétique (figure 3-2). Elle augmente de 42% avec les types d'applications et les arbres de décision (de 2,6 à 3,7) et de 73% avec les aperçus de patrons (de 2,6 à 4,5) par rapport au mode de référence. De plus, l'écart-type est plus faible pour les nouveaux modes, particulièrement pour les aperçus, que pour l'index alphabétique : les participants s'accordent mieux sur les notes qu'ils ont données aux nouveaux modes.

Les aperçus de patrons reçoivent les meilleures notes de satisfaction et sont le mode favori de sept participants sur 10, tout en étant le second mode le plus rapide. Inversement, l'index alphabétique de référence est le mode le plus rapide mais avec les pires notes de satisfaction. Les arbres de décision et les types d'application se situent entre les deux précédents en termes de vitesse et de satisfaction. Notons néanmoins quand les participants ont dû choisir leur mode de présentation favori, trois utilisateurs sur 10 se sont prononcés en faveur des types d'applications, alors qu'aucun ne se prononce en faveur des arbres de décision (annexe D.8).

¹² Parfois quatre dans le cas de l'index alphabétique.

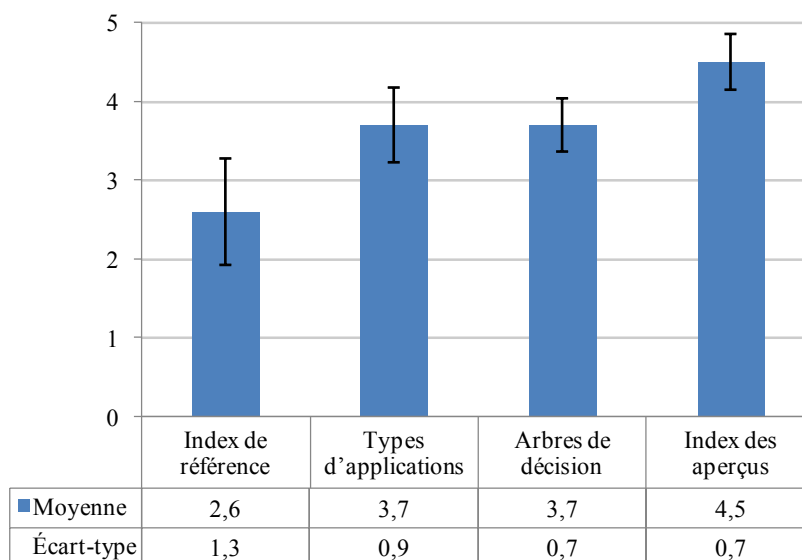


Figure 3-2: Satisfaction moyenne des utilisateurs

3.2.4 Pertinence des patrons de conception trouvés

Sur les 37 patrons trouvés, 35 sont appropriés à leurs interfaces respectives, et deux n'y sont pas applicables : les participants ont lu mais mal interprété les patrons ; ils les ont choisis en les pensant applicables alors qu'ils ne l'étaient pas.

La spécificité des 35 patrons restants varie en fonction des modes de présentation. Sur l'échelle de spécificité précédemment définie, le mode de référence obtient un score de 1,25. Les types d'applications et les arbres de décision obtiennent respectivement 1,9 et 2 et les aperçus terminent à 1,5. La note de pertinence finale combinant ces deux facteurs accentue la différence entre le mode de présentation de référence, qui descend à un, et les autres modes de présentation, qui restent à leurs valeurs précédentes. On obtient donc au final un gain de 89% pour les types d'applications (de 1 à 1,9), de 100% pour les arbres de décision (de 1 à 2) et de 50% pour les aperçus de patrons (de 1 à 1,5) comme montré figure 3-3.

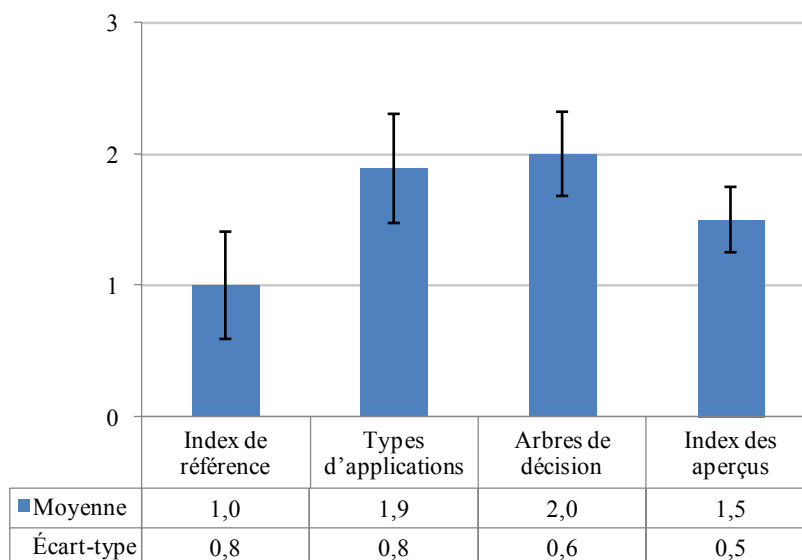


Figure 3-3: Pertinence moyenne des patrons trouvés

3.2.5 Corrélation des résultats expérimentaux avec les éléments fournis par le questionnaire biographique

Le questionnaire biographique confirme que tous les participants connaissaient les patrons de conception de programmation orientée-objet¹³, que ce soit via leur formation initiale ou leur emploi chez CAE, et qu'aucun d'entre eux n'avait entendu parler de patrons de conception d'interfaces utilisateur.

On a demandé aux participants au début du test d'estimer leur habileté à concevoir des interfaces. Cette valeur était relevée pour voir si un intérêt naturel pour les questions d'utilisabilité se traduisait par un biais favorable à l'usage des patrons.

Les résultats expérimentaux ne nous ont pas permis d'établir un lien entre l'habileté estimée des participants d'une part et leur performance ou leur préférence avec un mode de navigation en particulier d'autre part (annexes D.11 à D.13). De manière surprenante, il n'est pas non plus possible de dégager une préférence pour un mode de présentation, ni une performance sur un

¹³ L'ouvrage du « Gang of Four » de Gamma et al. qui a introduit et popularisé le concept des patrons de conception dans le monde du développement logiciel.

temps de recherche ou sur la pertinence d'un patron trouvé, en fonction d'un profil professionnel. Les préférences et les niveaux de performance des participants sont complètement indépendants des postes qu'ils occupent dans l'entreprise ; il n'est pas possible de distinguer des tendances dans les résultats en fonction des trois catégories de participants précédemment décrites (développeurs logiciels, gestionnaires d'équipe et autres).

Les données relevées ne sont pas suffisantes pour chercher d'autres explications à ces résultats. Il serait cependant intéressant de renouveler l'expérimentation en récoltant d'autres informations, parmi lesquelles leur besoin au moment de l'utilisation de la bibliothèque. Sont-ils en mode « résolution de problème » ou « dégustation de l'information » ?

Ces pistes de recherche expliqueraient peut-être les variations des résultats des participants.

CHAPITRE 4 DISCUSSION DES RÉSULTATS

Dans ce chapitre, les résultats précédents sont interprétés, puis des recommandations en sont tirées pour un usage immédiat en milieu professionnel. Nous y adjoignons quelques conseils émanant d'une année de pratique des patrons de conception dans une grande entreprise. Enfin, nous proposons des actions à mettre en œuvre pour terminer la mise en place de la bibliothèque de patrons chez CAE.

4.1 Interprétation des résultats

La réponse globale aux modes de présentation des patrons a été très positive et la bibliothèque de patrons d'interfaces utilisateur de CAE a été jugée attractive comme expliqué ci-dessous. Les participants, qui n'avaient pas de connaissance préalable des patrons de conception, ont été en mesure de trouver des patrons pertinents vis-à-vis de leur problématique, de le faire avec un taux de satisfaction élevé et dans un temps comparable à la norme, et ce quel que soit leur profil professionnel.

4.1.1 Temps requis pour trouver un patron

Le temps de recherche d'un patron approprié est plus important avec les nouveaux modes de présentation qu'avec le mode de référence.

Le contexte et le conseil fourni par les nouveaux modes de présentation peuvent demander plus de temps de traitement mais mènent vers des patrons plus pertinents et une meilleure sensibilisation aux questions d'utilisabilité. Du reste, la satisfaction des utilisateurs augmente légèrement plus vite que le temps requis pour trouver un patron (figure 4-1).

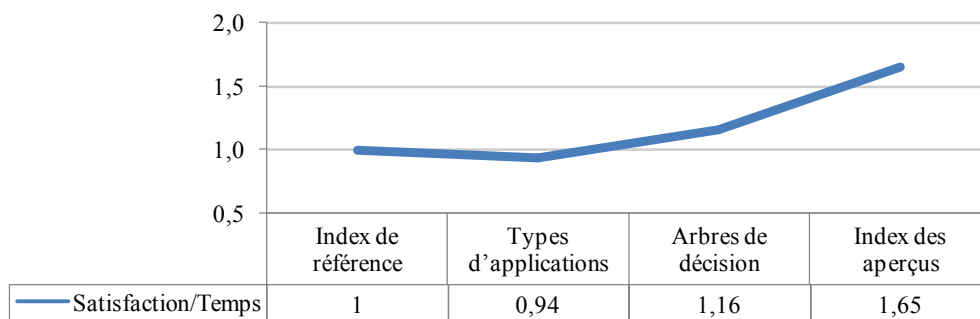


Figure 4-1: Rapport de la satisfaction des utilisateurs sur le temps de recherche des patrons

4.1.2 Arbres de décision et types d'application

Les types d'applications et les arbres de décision agissent comme des points d'entrée doubles et ciblés vers des sous-ensembles de la bibliothèque, la morcelant *de facto* en lots plus aisés à traiter pour les développeurs. Ce morcellement, ainsi que le contexte fourni par les descriptions textuelles et visuelles de ces modes, contribuent à construire un modèle mental des principes ergonomiques chez le développeur. Cela diminue également la charge de travail de l'utilisation de la bibliothèque, les développeurs n'ayant plus besoin de la connaître par cœur pour sélectionner le bon patron. Il leur suffit de reconnaître le problème d'utilisabilité auquel ils sont confrontés ou le type d'application sur lequel ils travaillent pour obtenir une solution.

La complémentarité des modes a également été remarquée par les participants. Les types d'applications ont reçu de meilleures notes de la part des participants impliqués dans des nouveaux projets dont l'interface n'était pas encore faite, parce qu'ils leur donnaient un point de départ. Au contraire, les arbres de décision ont reçu de meilleures notes de la part des participants travaillant sur des produits matures avec des problèmes d'utilisabilité identifiés.

Nous relèverons un dernier résultat : les patrons trouvés avec l'index alphabétique et les aperçus étaient génériques, incluant des recommandations tout-usage comme *Bonne typographie* ou *Bonne couleur*. Au contraire, les patrons trouvés avec les arbres de décision et les types d'applications, comme *Cape des ombres* ou *Méga menu-déroulant*, étaient plus spécifiques et donc à même d'amener plus de valeur lors de la conception. La liste des patrons trouvés est indiquée dans l'annexe D.5 ; leur pertinence dans l'annexe D.7.

4.1.3 Aperçus de patrons

Les aperçus ont été vraiment appréciés et semblent avoir servi à deux niveaux : d’abord comme une introduction à la bibliothèque, jouant sur la curiosité des participants qui s’amusaient à deviner le sens d’un patron à l’aide de son seul aperçu, même quand ils avaient déjà trouvés un patron *approprié*. Ensuite comme une référence rapide pour les utilisateurs avancés, à même de fournir de l’information riche d’un coup d’œil.

C’est pour ce mode de présentation que les participants ont eu les commentaires les plus positifs :

« Index visuel, pas trop d'informations mais juste ce qu'il faut. »

« C'est la plus simple pour browser puis explorer. »

« Non c'est bien ça, avec quoi tu fais ça ? C'est vraiment bien. »

« C'est génial, c'est simple, ça va à l'essentiel. »

« Ça l'index visuel c'est vraiment bien, les images sont très parlantes. »

« Index visuel m'a surpris, très intéressant. C'est une grande vitrine pour faire du lèche-vitrine, je clique par curiosité. »

Les aperçus de patrons sont également très pratiques pendant la rédaction de la bibliothèque. Les aperçus font office de *paramètres fictifs (placeholders)* pour remplir peu à peu de contenu et rédiger organiquement la bibliothèque, aidant le rédacteur à se faire une idée de sa taille finale et des relations entre patrons. Il est possible d’itérer sur le sens des patrons ou leur catégorisation plus facilement qu’avec des versions complètement rédigées qu’il faudrait remanier à chaque ajustement. Les aperçus permettent aussi de fournir très rapidement un échantillon des patrons aux utilisateurs de la bibliothèque, leur permettant d’en juger l’envergure prévue, de donner un premier feedback voire d’exploiter les aperçus comme des patrons rudimentaires.

Cependant, il serait néfaste de se concentrer uniquement sur les aperçus. Les arbres de décision et les types d’applications, bien que moins ostensibles, fournissent des réponses plus précises et amènent aussi ce dont les développeurs manquent le plus, du contexte. Les développeurs logiciels peuvent ainsi se sensibiliser peu à peu à l’existence et à l’origine des problèmes d’utilisabilité, et aux raisonnements menant à leur résolution.

4.1.4 Rédaction adaptée aux développeurs

En l'absence de mesure standard dans l'industrie, l'évaluation des patrons se borne pour le moment à des anecdotes (Vlissides, 1997), rendant la comparaison entre différentes rédactions difficile. Le style d'écriture des patrons CAE n'a donc pas été évalué de manière formelle dans le cadre de ce mémoire, nous nous sommes satisfaits de sa validité apparente.

La plupart des participants ont toutefois spontanément loué le choix, la quantité et la qualité des exemples ainsi que la rédaction sous forme de liste à points, alors que ces aspects n'avaient pas été mentionnés par l'examineur :

« C'est très riche comme contenu, beaucoup d'exemples, c'est bien. C'est difficile de diverger du patron à partir des exemples, l'utilisateur comprend bien la directive, l'image mentale, j'aime particulièrement ça, c'est très pertinent. Ya pas de texte et on voit ce dont ça parle. C'est impressionnant. »

« Globalement c'est intéressant. Ça permet de trouver une terminologie déterminée pour des choses abstraites et difficiles à décrire. »

L'entrevue de fin de session était l'occasion de demander aux participants d'interpréter les patrons qu'ils avaient choisis. L'examineur a ainsi pu vérifier la justesse de leur interprétation. Plusieurs participants ont cependant indiqué à cette occasion qu'ils trouvaient les patrons ou les modes de présentation trop verbeux ; c'est un point qu'il faudra améliorer ultérieurement.

4.2 Recommandations

Les recommandations suivantes sont déduites de l'analyse précédente. Étant directement issues de notre travail chez CAE, elles s'appliquent préférentiellement à des bibliothèques de grande ampleur, pour des utilisateurs peu ou pas formés en utilisabilité ; elles peuvent cependant être utilisées avec profit dans une multitude de cas.

4.2.1 Commencer chaque patron par son aperçu

C'est une manière rapide et peu coûteuse d'améliorer la découvrabilité et la compréhensibilité des patrons pour les utilisateurs novices. Ce mode de présentation permet aux utilisateurs experts

de naviguer dans la bibliothèque en mode reconnaissance plutôt que rappel. C'est en outre une aide précieuse à la rédaction comme expliqué au chapitre précédent.

Les visuels employés dans les aperçus devraient être des schémas indépendants de toute plateforme technologique plutôt que des captures d'écran. En effet, les patrons sont intrinsèquement découplés du code et de l'apparence graphique des solutions qu'ils distillent ; associer de manière forte un visuel, donc une implémentation en particulier, à un patron, c'est compromettre précisément les éléments invariants que l'on essaie de communiquer. Les utilisateurs de la bibliothèque risquent en effet de court-circuiter la lecture du patron en se contentant de l'exemple donné en aperçu, et de n'implémenter que des solutions superficielles ou hors-contexte.

4.2.2 Supporter la bibliothèque dès que possible avec des types d'applications et des arbres de décision

Une fois que la bibliothèque est un peu plus stable et que le nombre de patrons dépasse la vingtaine, il devient intéressant d'introduire des arbres de décision et des types d'applications, qui conduisent à de meilleurs résultats que l'index alphabétique ou les aperçus seuls.

4.2.2.1 Types d'applications

Les types d'applications donnent développeurs logiciels ou aux responsables de projet un point de départ lors de la conception d'un nouveau produit ou d'un nouvel écran.

Ils disposent au moins des éléments suivants : un titre, une image représentative, une série de produits internes ou externes à l'entreprise ayant valeur d'exemple, une description en quelques phrases et enfin une liste à points des patrons proposés pour ce type d'application.

4.2.2.2 Arbres de décision

Les arbres de décision fournissent un patron en particulier pour résoudre un problème précis auquel est confronté l'utilisateur. Ils peuvent être représentés sous forme d'assistants interactifs ou de diagrammes (voir la section *2.5.1 Arbres de décision* pour plus de détails).

Les novices préfèrent les assistants interactifs qui les encadrent et les guident progressivement vers un résultat. Inversement, ils sont déroutés par les diagrammes, ne s'attendent pas à trouver

ce genre de contenu sur un site Web¹⁴ et sont submergés par la complexité de toutes les options présentées en même temps. Cette même complexité plaît à certains utilisateurs avancés qui apprécient pouvoir scanner toutes les options d'un coup d'œil et sauter directement à un patron sans avoir besoin de cliquer à plusieurs reprises plusieurs questions/réponses.

Dans les deux cas, l'arbre est présenté sous une forme significativement différente du reste du contenu présent dans la bibliothèque : visuels et boutons contre titres, paragraphes, liens hypertextes... Il convient donc de porter attention à l'utilisabilité de l'artefact en lui-même, pour que des utilisateurs puissent s'en servir sans assistance.

4.2.3 Garder un index alphabétique disponible

Même s'ils ne s'en servent pas, les utilisateurs sont rassurés par la présence d'une solution simple et sans surprise. L'index sert aussi parfois à jauger d'un coup d'œil la quantité de patrons disponibles.

Les conseils qui suivent ne découlent pas directement des résultats de nos évaluations, mais d'une année de pratique des patrons de conception en milieu professionnel.

4.2.4 Écrire pour un public néophyte

Le public principal (du moins en volume) de la bibliothèque est constitués de développeurs logiciels. Les considérations d'utilisabilité d'une interface, et donc l'utilisation de la bibliothèque, sont pour eux des tâches de soutien, annexes de leur occupation principale. Comme ils n'ont pas le temps ou la formation pour interpréter des textes longs et riches, il faut leur fournir un contenu approprié à une consommation rapide et ponctuelle, avec un besoin de réflexion minimal. Il est donc recommandé de rédiger de manière concise et prescriptive, en paragraphes courts, listes à points ou instructions séquentielles numérotées, à la manière des procédures. On se reportera avec profit à Wieringa (1999) pour des aides à la rédaction.

La rédaction devrait se conformer à la terminologie des développeurs et limiter au maximum le jargon spécifique au domaine des IHM. Les mots-clés seront définis dans un lexique pour établir et communiquer le consensus.

¹⁴ Rappelons à toutes fins utiles que la bibliothèque de patron est hébergée sur l'intranet de l'entreprise.

En outre, il est judicieux d'ajouter à chaque patron plusieurs exemples soigneusement choisis, car ils rendent les patrons plus faciles à comprendre et fournissent un éventail d'implémentations. Les exemples devraient être :

- Tirés d'au moins cinq plates-formes différentes, parmi les trois grandes familles de systèmes d'exploitation (Windows, Mac OS, Linux), des sites ou applications Web, des produits tactiles comme iOS, Android, webOS...
- Traités pour que la partie pertinente de la capture d'écran soit mise en valeur (par un encadrement ou par l'assombrissement du reste de l'image). Il n'est pas judicieux de rogner l'image pour n'en montrer que la partie pertinente car on perd alors beaucoup de contexte.
- Commentés, expliquant l'interaction présentée ou introduisant le comportement de l'utilisateur final. Répéter si besoin des informations présentes ailleurs dans le patron.
- Appropriés esthétiquement et de manière générale dépourvus de toute mauvaise pratique, même sur des parties « non-significatives » de l'image.

Il est également conseillé de fournir des aides à l'écran, au moins lors des premières utilisations de la bibliothèque. Ces aides sont généralement affichées comme des bandeaux ou des ballons pouvant être fermés ou rappelés par l'utilisateur à sa guise. La teneur de ces instructions d'usage est susceptible de varier au cas par cas, mais quelques éléments reviennent fréquemment :

- Une apostrophe expliquant que tous les développeurs sont concernés par la bibliothèque et qu'elle leur est destinée.
- Une présentation des patrons en quelques mots, expliquant qu'un même patron peut intervenir à plusieurs niveaux dans une interface, avec éventuellement un parallèle vers les patrons de Gamma et un lien vers une explication plus en profondeur.
- Une liste des avantages à utiliser les patrons (gain de temps, meilleur score lors de l'assurance qualité...).

Enfin, il convient de ne pas hésiter à faire des liens depuis les patrons vers les lignes directrices et vice-versa. Les développeurs qui viennent chercher de l'information dans la bibliothèque n'ont pas à être handicapés par le cloisonnement arbitraire que les ergonomes font avec leurs artefacts

de gestion de la connaissance ; bien souvent l'information cherchée est un mélange de patrons et de lignes directrices.

4.2.5 Désigner un rédacteur en chef

Les bons patrons sont difficiles et longs à écrire. Désigner un rédacteur en chef similaire au *pattern mentor* proposé par Beck et al. (1996) ou au *pattern shepherd* de Dearden & Finlay (2006) assure deux fonctions : rédiger la bibliothèque en profitant de la diminution de l'effort de rédaction qui vient avec l'expérience et gérer la communauté de développeurs.

4.2.5.1 Rédiger et maintenir la bibliothèque

Le rédacteur en chef doit piloter l'effort de mise en place et de maintenance de la bibliothèque, tel que décrit au chapitre Développement de la bibliothèque de patrons. Cela signifie faire une veille technologique, identifier et extraire les patrons, les rédiger, s'occuper du système de gestion de contenu, faire le lien avec les autres acteurs du processus, etc. gérer le projet de manière générale. Deux tâches en particulier ressortent car il est difficile de les impartir : le maintien du système de valeur du langage de patrons et la surveillance de la cohérence.

Nous savons déjà que le système de valeur est important : c'est le cadre intellectuel définissant la nature et la croissance d'un langage de patrons. La cohérence est une notion voisine, elle pourrait être définie comme la capacité des patrons à être reliés les uns aux autres au sein d'une même bibliothèque sans interstice ni recouvrement :

« Un bon langage (...) est morphologiquement complet quand les patrons forment ensemble une structure complète, pleine et sans vide » Traduction libre du texte d'Alexander (1979).

Quand elle se dégrade, il faut réajuster les patrons les uns aux autres pour qu'ils soient à nouveaux pertinents et applicables de concert. C'est le phénomène qu'évoque Welie (2006) dans son mémoire :

« Une fois qu'une collection commence à contenir, mettons, plus que 50 patrons, on découvre qu'il faut ajuster ces patrons à nouveau et affiner l'énoncé du problème et du contexte pour que le patron retrouve sa pleine signification » (traduction libre du texte).

La cohérence d'une bibliothèque partage avec le système de valeur le fait d'être à la fois séminal et très abstrait : leur existence et leur qualité sont critiques pour la performance de la bibliothèque

(voir la section *1.1.3 Les patrons de conception*), mais leur dégradation est lente et insidieuse. Les notions de cohérence et de résilience sont à ce jour mal connues et mériteraient une étude approfondie, mais il est certain que leur surveillance et leur maintien (proactif ou correctif) exigent rigueur et vigilance, ainsi qu'un suivi dans le temps et une compréhension fine du système. C'est une charge mentale importante, difficile à impartir, et qui échoit naturellement à l'éditeur de la bibliothèque car c'est lui qui a la vision du haut vers le bas du système nécessaire.

4.2.5.2 Faire le lien avec les développeurs

Le rédacteur en chef fait office de figure publique de la bibliothèque de patron. C'est lui qui en est responsable aux yeux des développeurs et répond à leurs questions concernant la bibliothèque, comme par exemple chez CAE :

- Quel département en a la responsabilité ? Qui va payer (en budget, en temps) pour ça ?
- Qui pilote le projet, assure le suivi ? Est-ce que ça va changer mes méthodes de travail ? Est-ce obligatoire ?
- Est-ce que ça vaut le coup d'investir là-dedans, est-ce que l'initiative a du soutien ?

Toujours d'après Dearden & Finlay (2006), un rédacteur en chef peut aider à faire le compromis entre encourager des bonnes pratiques de conception basées sur les patrons et décourager des attentes disproportionnées des développeurs logiciels quant aux patrons. Le rédacteur peut aider les développeurs à reconnaître les patrons qu'ils utilisent déjà et comment ils pourraient les réutiliser dans des projets ultérieurs. Il s'assure aussi que les développeurs n'appliquent pas toujours les mêmes patrons quelle que soit leur pertinence par rapport au problème¹⁵.

Enfin, le rédacteur en chef anime la communauté des développeurs, répond à leurs demandes de support et reçoit leur feedback.

Avoir un tel référent rassure les développeurs et diminue leur résistance au changement.

¹⁵ Syndrome « Quand on a qu'un marteau, tout ressemble à un clou ».

4.2.6 Mettre de côté des ressources pour « l'utilisabilité de l'utilisabilité »

Comme expliqué dans la section 2.5.4 *Vérification et validation*, il est judicieux de mettre de côté une partie des ressources dévolues aux patrons pour l'allouer à la rédaction des modes de présentation. Ceux-ci vont en effet valoriser l'investissement que représente la bibliothèque de patrons en la rendant accessible à un public plus large.

4.3 Actions à prendre

Cette section présente les trois efforts à mener en parallèle pour assurer la qualité, l'usage et la pérennité de la bibliothèque de patrons de CAE. Les deux premiers axes de travail, respectivement l'augmentation de la quantité de patrons présents dans la bibliothèque et la gestion sociale de l'acceptation des patrons, pourraient s'appliquer avec profit à la plupart des bibliothèques.

Le troisième effort en revanche est plus spécifique aux grandes organisations techniques puisqu'il concerne le développement des modes de présentation à destination des développeurs.

4.3.1 Continuer à alimenter la bibliothèque en patrons

La bibliothèque de CAE contient pour le moment 30 patrons sur un total estimé de 200. La participation des développeurs est indispensable à l'écriture des patrons restants car ce sont eux qui vont indiquer les patrons dont ils ont le plus besoin et qui vont fournir des éléments de *Problème* et de *Contexte* pour ces patrons. Un bon moyen d'impliquer les développeurs dans la rédaction des patrons serait de mener des ateliers de remue-méninge ponctuels :

« Ce modèle rassemble régulièrement les membres d'une équipe pour travailler sur une poignée de problèmes à la fois. Ils écrivent en collaboration et discutent des problèmes et des solutions. À la fin des deux heures, ils ont documenté l'essentiel de quelques patrons. Un bénéfice supplémentaire est que l'équipe a un consensus sur l'approche et la solution. » Traduction libre du texte de Malone (2010).

Ainsi, les développeurs peuvent participer sans trop perturber leur travail car leur participation est restreinte dans le temps, plutôt que diffuse et en parallèle de leurs tâches principales. Un autre avantage de cette méthode est qu'elle instaure un filtre *de facto*, les idées étant triées par le rédacteur en chef avant d'être publiées.

L'expertise du rédacteur en chef assure ainsi que les suggestions et demandes des développeurs soient traduites en patrons efficaces, communiquant ce qui est bon plutôt que ce qui est commun, se basant sur des invariants plutôt que sur des implémentations, tout en respectant l'intégrité du système de valeur et du principe structurant de la bibliothèque de manière à bénéficier des outils déjà mis en place.

Les spécialistes en IHM collaboreront eux via un *processus de validation*, c'est-à-dire une revue collaborative des brouillons de patrons. Ce processus sert à la fois à identifier les manques ou faiblesses dans les brouillons et à construire le consensus au sein de l'équipe.

En synthétisant les contributions des développeurs et de ses collègues en IHM, le rédacteur en chef est en mesure de rédiger des patrons qui répondent aux besoins réels des utilisateurs de la bibliothèque et d'ordonnancer la rédaction pour parer au plus urgent.

4.3.2 Évangéliser et rassurer

Toute production de nouvelle technologie s'accompagne de résistances et d'oppositions sociales comme l'explique Lejeune (2008) :

« Le succès d'une technologie ne dépend pas seulement de la rationalité technique, mais également de facteurs strictement reliés aux comportements des humains, à leur culture, leur système politique (...) La diffusion de la technologie, comme sa production, est tout aussi socialement déterminée : la société, notamment à travers l'action des utilisateurs, détermine la façon dont la technologie se diffuse. »

La diffusion d'une nouvelle méthode de travail rencontre exactement les mêmes obstacles, a fortiori dans notre cas où elle est exogène au corps de métier considéré. Nous verrons donc dans cette section pourquoi considérer l'aspect social des patrons de conception et comment gérer au mieux leur diffusion dans le milieu professionnel des développeurs logiciels.

4.3.2.1 Prendre en compte l'aspect social

Les résistances et oppositions lors de la diffusion d'une nouvelle méthode varient du refus au sabotage en passant par l'adoption d'attitudes *faux-participatives*. Les conséquences potentielles de ces frictions peuvent être relativement bénignes, comme une diminution temporaire de la productivité, ou plus grave, comme un échec de l'implémentation de la nouvelle méthode, entraînant à son tour un handicap compétitif majeur. Le problème est amplifié dans le cas des

patrons de conception car la collaboration du personnel est requise à deux moments : en amont pour fournir la matière première des patrons, et en aval pour les utiliser.

La collaboration en amont est d'autant plus importante que la connaissance de l'entreprise est en grande partie implicite et la capturer requiert une participation de bonne volonté de la part des développeurs. La collaboration en aval conditionne directement l'utilité de la bibliothèque de patrons car la technologie devient performante quand on a choisi de se l'approprier (Lejeune, 2008).

4.3.2.2 Pistes pour évangéliser et rassurer

Il est judicieux de prendre des mesures préventives pour gérer l'acceptation sociale des patrons de conception. La gestion du changement la plus élémentaire consiste à aviser les utilisateurs de la disponibilité de la bibliothèque, à en décrire la nature et expliquer pourquoi et comment s'en servir en mettant en valeur l'importance des considérations d'utilisabilité, la facilité d'utilisation des patrons ou le gain de productivité de la réutilisation de code.

Les évaluations empiriques ont en outre fait ressortir l'inquiétude des développeurs, qui ont alors tendance à lire le patron en diagonale sans rien en retirer. Il convient de les rassurer sur les points suivants pour les encourager à participer :

- La bibliothèque est un effort dans la durée, les ressources investies ne vont pas partir en fumée. Il existe un responsable à qui s'adresser directement.
- Les patrons ne sont pas imposés au nom d'un dogme¹⁶ mais exposent des éléments de réflexion pour que les développeurs puissent faire leur propre choix (l'élargissement du travail est vécu positivement).
- Les patrons ne sont pas une manière déguisée de demander subitement aux développeurs de refaire toutes leurs interfaces, et ne modifieront pas leurs habitudes de travail ou la dynamique organisationnelle.
- Le code lié à chaque patron est à jour, lisible et maintenable ; il est généralement fourni par un collègue.

¹⁶ Plusieurs testeurs ont cité ce mot (« dogme ») en particulier. Lors des tests informels, les gens ne se contentaient pas de la solution ex machina et demandaient à voir la section « Raisonnement », alors masquée par défaut.

Ce travail de communication est appelé *évangélisation*. Il est beaucoup plus efficace si la hiérarchie participe en tant que relais de promotion, par exemple via le comité des architectes précédemment mentionné.

Dans tous les cas, il est important de rédiger et diffuser des dépliants de type « Le B.a.-ba des patrons » couvrant les points précédents et d'autres, plus en profondeur, comme les moments recommandés pour utiliser les patrons dans un processus de développement logiciel, la manière pratique d'instancier ou de combiner des patrons dans une conception, les modalités de participation ou de feedback... Les développeurs logiciels peuvent ainsi s'approprier la méthode à leur rythme, en expérimentant par eux-mêmes.

Enfin, parce que la hiérarchie est d'accord et que l'organisation s'y prête, l'implémentation de mécaniques de jeu¹⁷ est envisagée chez CAE pour promouvoir l'existence et l'utilisation de la bibliothèque de patrons. Un système interne de *succès*, comparable à ce qui se fait dans certains réseaux sociaux ou jeux vidéo, pourrait augmenter la notoriété et l'attractivité de la bibliothèque (Watters, 2010; Hamari & Eranti, 2011) et encourager l'application de patrons non fonctionnels.

4.3.3 Corriger et enrichir les méthodes mises en place

Cette section expose les faiblesses que les retours utilisateurs ont révélées dans nos propositions. Des moyens pour enrichir les innovations proposées en les rendant utiles à plus de gens et dans plus de cas d'usages sont étudiés en parallèle. Le tout est combiné en pistes de réflexion holistiques à l'usage de travaux ultérieurs.

4.3.3.1 Travailler l'utilisabilité de la bibliothèque

La bibliothèque de patrons de conception de CAE est hébergée sur l'intranet de CAE par l'intermédiaire d'un système de gestion de contenu, tel qu'expliqué dans la section 2.4.3 *Développement du système de gestion de contenu*. La problématique de l'utilisation des patrons par les développeurs est donc autant un problème d'accessibilité des artefacts du domaine de l'ergonomie pour les développeurs logiciels qu'un problème conventionnel d'utilisabilité de site Web.

¹⁷ Juste contrepartie à la mécanique de punition de l'assurance qualité, expliquée ci-après.

Cette dichotomie se retrouve dans les faits, les commentaires des participants lors des tests portant indistinctement autant sur les modes de navigation, les patrons en eux-mêmes que sur le site de la bibliothèque, comme par exemple : « Le fil d'Ariane ressemble à des onglets, le soulignement ressemble à une ombre. L'espace est trop grand, on dirait des options séparées et pas un chemin ».

Améliorer le site intranet de la bibliothèque, en application directe de notre propre recommandation de « l'utilisabilité de l'utilisabilité », devrait donc conduire à des gains faciles¹⁸ dans le taux de participation et la performance des utilisateurs (pertinence des patrons trouvés, temps de recherche, etc.). Ces ajouts ou corrections vont du très large au très précis mais sont tous implémentables à court terme. En voici quelques exemples à titre indicatif :

- De manière générale le site profiterait d'un meilleur guidage, ne dévoilant ses possibilités que graduellement aux utilisateurs novices. Ne pas afficher tous les modes de navigation tout de suite, les montrer un par un avec un visuel simplifié mettant en avant le cas d'usage typique et les conséquences (profitables) de son utilisation. Penser aussi à répondre en quelques mots à la question *Qu'est ce qu'il y a pour moi là-dedans ?* que se posent les développeurs logiciels.
- Donner des titres à chaque fragment de contenu. Par exemple précéder la liste des patrons dans un type d'application par un titre *Patrons proposés*, d'une part pour aider la lisibilité de la page, d'autre part pour expliciter la nature d'une telle liste de lien.
- Mettre les assistants des arbres de décision plus en valeur que les diagrammes dans un même arbre de décision, et bien préciser qu'on trouve exactement le même contenu dans les deux modes de présentation pour que les utilisateurs qui ne trouvent pas leur bonheur avec un mode puissent essayer avec l'autre.
- Diminuer la taille des illustrations dans les aperçus de patron de manière à pouvoir mettre plus d'aperçus par page.
- Renforcer l'affordance des liens dans l'index des aperçus.

¹⁸ On qualifie ces gains de « faciles » car le sujet du design Web est bien connu et dispose de nombreux outils. On dispose de beaucoup moins de ressources sur les patrons de conception en général et les modes de navigation en particulier.

4.3.3.2 Supporter la granularité des patrons de manière plus formelle

De manière générale, les développeurs logiciels comprennent l'aspect *fractal*, c'est-à-dire invariant et applicable à plusieurs échelles, des patrons. Ils finissent cependant par n'instancier qu'une occurrence du patron à la fois, généralement celle correspondant à l'exemple qui leur a le plus plu, après quoi ils pensent « en avoir fini » avec le patron.

La fourniture d'exemples schématisés, à l'instar des aperçus de patrons, la création de *sous-patrons* raffinant un patron parent ou d'autres méthodes à déterminer pourraient aider les développeurs à comprendre qu'il est à la fois possible et bénéfique d'instancier un patron unique en plusieurs composants sur le même écran si l'occasion s'y prête.

4.3.3.3 Supporter d'autres profils d'utilisateurs

Bien que la bibliothèque de patrons de CAE soit accessible librement à tous les employés, elle ne supporte activement que des sous-ensembles prédéterminés du personnel : les développeurs et architectes logiciels, les chefs d'équipe de développement logiciel et les ergonomes.

Il serait très intéressant de soutenir d'autres profils au sein de l'entreprise : des responsables produits (à la formation plus commerciale que technique), des échelons plus élevés de la hiérarchie, mais aussi les stagiaires, consultants, développeurs juniors... Chacun pour des raisons différentes. Ainsi les cadres supérieurs seraient informés du consensus et de l'état de l'art au sein de l'entreprise. La communication avec le client pourrait se faire via un vocabulaire unique, contrôlé et illustré. Les stagiaires, consultants ou développeurs juniors pourraient eux s'intégrer très vite dans l'entreprise en lisant la documentation que forment les patrons de conception en complément de leur appairage avec un employé-tuteur. Tous pourraient profiter de l'effet de *langue franche* des patrons de conception, se plaçant alors dans une dynamique de rendements croissants d'adoption et d'économies d'échelle.

Concrètement, supporter ces utilisateurs implique d’adapter le discours d’évangélisation et surtout de développer de nouveaux modes de présentation¹⁹ prenant en compte la culture et le mode de travail de ces profils.

4.3.4 Transformer la bibliothèque en langage

Nous savons que l’utilité des patrons de conception croît grandement avec leur organisation en langage car «un langage de patrons rend le design génératif possible et contribue significativement à l’intégrité conceptuelle du produit final » (Mahemoff & Johnston, 2001).

Pour le moment les patrons de CAE ne forment qu’une bibliothèque ; ils disposent certes d’un système de valeur rudimentaire mais ne bénéficient pas d’un principe structurant (figure 4-2). Cette section présente les axes de travail que nous envisageons pour y remédier

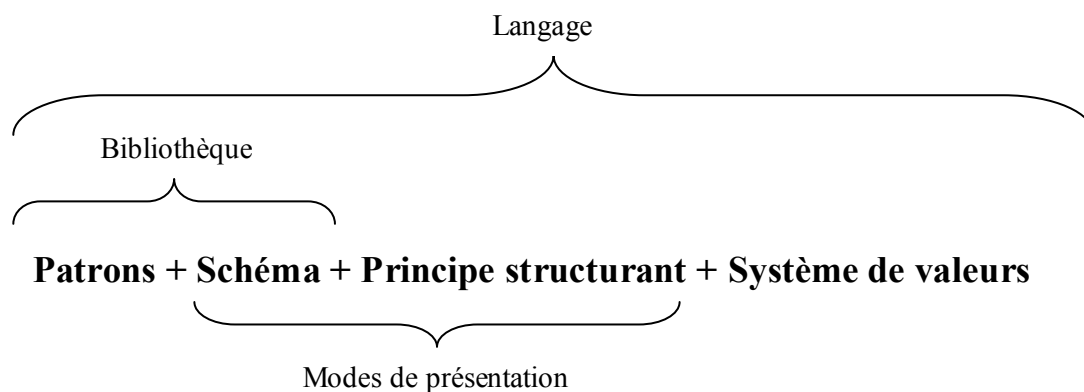


Figure 4-2: Composantes d'une bibliothèque et d'un langage de patrons

¹⁹ On pourrait ajouter un catalogue des modes de présentation à la liste de la documentation manquante pour la recherche ou la rédaction sur les patrons de conception. Un tel recensement constituerait un sujet de recherche à part entière.

4.3.4.1 Les types d'applications comme langage

À ce jour, nos patrons sont naturellement structurés autour des arbres de décision et des types d'applications qui agissent comme des centres thématiques. Les arbres de décision ne pourraient pas remplir la fonction générative du langage car ils ne fournissent finalement qu'un patron.

Les types d'applications fournissent en revanche un sous-ensemble cohérent et ciblé de solutions dont la combinaison est plus grande que la somme des parties, ils se rapprochent conceptuellement beaucoup plus de la notion de langage de patrons de conception. Nos types d'applications sont pour le moment des idiomes, des dialectes plutôt qu'un langage : ils forment des centres thématiques finis et relativement isolés les uns des autres quand il faudrait un maillage continu de solutions, et sont surtout relativement « plats » et de bas niveau, se concentrant sur l'adéquation et la qualité des composants pour un écran donné mais ignorant en grande partie les problématiques d'architecture de l'information, de flux de tâches de l'utilisateur ou de mise en page sur un écran.

Plusieurs éléments pourraient contribuer à nous rapprocher d'un véritable langage. Premièrement rédiger les types d'applications non plus de manière tolérante mais sous le formalisme des patrons de conception en ferait naturellement des patrons de haut niveau chapeautant le corpus existant. Deuxièmement, prendre en compte les problématiques précédemment évoquées, le *quand* et le *où* autant que le *quoi* et le *comment* afin de supporter la conception de manière plus large. Troisièmement, porter une attention accrue aux objectifs d'affaires ou à d'autres perspectives similaires. Même si les patrons de conception devaient rester indépendants d'une stratégie commerciale comme ils sont indépendants d'une plate-forme technologique ou d'un style esthétique, l'étendue des éléments à considérer augmente au fur et à mesure que l'on s'éloigne de l'exécution d'un produit pour se rapprocher de sa définition. Les patrons de haut niveau devraient prendre en compte l'aspect pluridisciplinaire (commercial, technique, politique) de la définition du produit pour que l'équipe d'ingénierie des facteurs humains soit à même de collaborer avec les autres parties impliquées à ce stade de décision.

4.3.4.2 Mieux intégrer les patrons dans les différents processus de l'entreprise

Rappelons que les patrons de CAE seront utilisés d'abord par les développeurs logiciels, comme source ponctuelle de conseils ; il ne serait donc pas judicieux d'établir un processus de conception complet centré sur les patrons comme PSA (Granlund et al., 2001), POD (Javahery &

Seffah, 2002), Moudil (Gaffar, Sinnig, Javahery & Seffah, 2003) ou PCB (Sinnig, Javahery, Forbrig & Seffah, 2005). Nous avons choisi de livrer dans un premier temps la bibliothèque en accès libre²⁰, mais cette approche est un pis-aller puisqu'elle demande un effort de la part des développeurs : il faut qu'ils prennent conscience du besoin d'assistance lors de la programmation d'interface puis qu'ils aillent tirer l'information depuis la bibliothèque.

Un meilleur moyen d'assurer l'usage et l'utilité des patrons serait de les tresser dans les processus que les développeurs logiciels utilisent au jour le jour ; une telle intégration permettrait de *pousser* la bibliothèque vers les développeurs au moment opportun, ledit moment variant en fonction du processus considéré.

Pour les processus comme *Agile* ou *Scrum*, les patrons interviendraient lors de la définition et de l'implémentation effective des carnets de tâches (*backlogs*). Agile et Scrum fonctionnent en cycles itératifs de deux à quatre semaines. Les tâches à effectuer par les équipes de développement pendant ces cycles sont énumérées dans des carnets de produits (*products backlogs*) et des carnets d'itération (*sprints backlogs*), et sont convertissables assez directement en fonctions ou en groupes de fonctions correspondant à leur tour à des patrons. Par exemple, si le carnet de produits demande une fonction de recherche, alors le carnet d'itération pourrait traduire ce besoin en trois éléments : *Création d'un index*, *Implémentation d'un moteur de recherche tierce-partie* et *Codage d'un GUI pour la recherche*. La liste de vérification du développeur lui indiquerait à ce moment de chercher des solutions dans la bibliothèque de patrons et il pourrait y trouver les patrons *Recherche libre*, *Recherche à facettes*, *Recherche avancée*, *Complétion* et *Suggestion automatiques*, etc. pour l'aider à implémenter la recherche sans tout commencer de zéro.

Le principe fondamental de ces méthodes est l'ajout de fonctions multiples et complémentaires, chacune d'ampleur limitée pour qu'elles soient opérationnelles avant la fin du cycle. Ce mode opératoire établit une passerelle forte avec les patrons de conception : non seulement la bibliothèque est un réservoir à connaissance tout-usage facile à consommer pour les équipes de développement en mode itératif, mais en plus la forme que prend ce savoir, des entités finies, granulaires et auto-suffisantes, reflète la forme des artefacts élémentaires de Agile ou Scrum.

²⁰ Eu égard à notre agenda serré autant que des convictions exposées dans la section éponyme.

Pour d'autres processus de développement logiciel comme ISO 9241 ou ISO 13407 le lien avec une bibliothèque de patrons est encore plus naturel puisqu'une étape de la norme demande explicitement de « Fournir des solutions de conception » (figure 4-3).

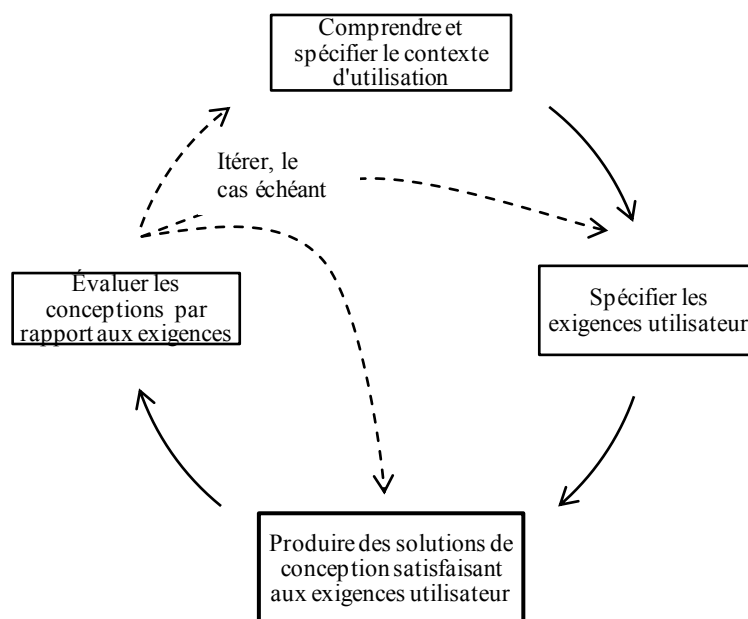


Figure 4-3: Interdépendance des activités de conception centrée sur l'opérateur humain

Adapté de ISO 9241 (ISO, 2010).

Enfin, le lien des patrons de conception avec le processus d'assurance qualité d'une entreprise est le plus prometteur. L'assurance qualité consiste à évaluer la conformité d'un produit à ses spécifications, et le cas échéant à entreprendre des actions correctives ; c'est un processus fondamental, répandu et accepté dans l'organisation. Dans une entreprise comme CAE, les produits sont évalués à l'issue de chaque cycle de conception (ou *sprint*) par une équipe dédiée, qui compare le logiciel à une liste, présentée sous forme de questions, de performances et de comportements attendus.

Les patrons de conception peuvent intervenir à deux niveaux dans l'assurance qualité : d'une part comme matériel de référence lors de la rédaction des exigences ou comportements attendus, fournissant la compréhension fine du problème et du contexte nécessaire à l'établissement d'une

grille pertinente et graduelle de notation, d'autre part comme suggestion de correction associée à une question, dans le cas où le logiciel ne satisferait pas cette exigence en particulier (dans les faits on associe soit un patron en particulier, soit un arbre de décision). Pouvoir lier de manière aussi précise un patron à une exigence en particulier est très intéressant car cela assure son utilisation, les développeurs logiciels étant assujettis à l'examen de l'assurance qualité. Cela est d'autant plus vrai pour les requis non-fonctionnels, souvent parents pauvres des carnets de tâches des équipes de développement mais bien présents dans le modèle d'assurance qualité.

L'inconvénient de cette méthode est qu'elle ne fait intervenir les patrons qu'après le processus de conception, les suggérant *de facto* pour l'itération suivante. Dans tous les cas, l'intégration des patrons avec le processus d'assurance qualité est déjà en cours chez CAE et constitue un effort prioritaire de l'équipe des facteurs humains.

Une connexion efficace de la bibliothèque à un ou plusieurs des processus évoqués exige du travail et une grande connaissance de la culture et du fonctionnement de l'entreprise pour atteindre l'ampleur, la granularité et la forme technique appropriée.

CONCLUSION

La conception des logiciels de contrôle de simulateurs de vol est une activité coûteuse dont la complexité croît avec les innovations technologiques et les exigences des clients. Plusieurs corps de métiers y participent : des experts des facteurs humains mais aussi des développeurs logiciels qui n'ont pas été formés en ergonomie. Les patrons de conception d'interfaces utilisateur, stockant la connaissance de manière formelle et pédagogique en vue de sa réutilisation, peuvent améliorer la qualité et la productivité du processus de conception. Ces avantages ont été reconnus par des géants du secteur de l'informatique qui ont adopté la méthode ; ils seraient un bon moyen de capitaliser sur les efforts réalisés par CAE.

Les patrons sont cependant rédigés par et pour des experts en ingénierie des facteurs humains, et plusieurs études montrent que cela exclut de fait les développeurs logiciels. Ces derniers ont du mal à naviguer dans des bibliothèques de patrons, et notamment à trouver des patrons applicables à leurs problèmes, les comparer ou identifier parmi eux les plus pertinents et ceux qui peuvent être utilisés de concert.

Nous nous sommes donc efforcés dans ce mémoire de mettre au point des modes de présentation des patrons d'une bibliothèque de sorte qu'ils soient accessibles autant aux développeurs logiciels qu'aux ergonomes.

Nous avons commencé par une revue de littérature établissant le cadre d'usage des patrons de conception par rapport aux autres artefacts de gestion de la connaissance en usage dans l'industrie, les lignes directrices et les composants. Nous avons ensuite identifié et étudié l'état de l'art des bibliothèques de patrons les plus populaires, en portant une attention particulière à la manière dont leurs patrons étaient rédigés et présentés aux lecteurs. Ce tour d'horizon nous a fourni un point de départ informé pour développer la bibliothèque de patrons de CAE. Nous l'avons complété par des inspections heuristiques des interfaces des produits CAE, une lecture de l'outil de suivi de bogues de l'entreprise et des entretiens avec les quatre membres de l'équipe d'ingénierie des facteurs humains et une trentaine de développeurs logiciels. Le matériel ainsi relevé a été rédigé en une trentaine de patrons avec force liste à points et exemples. Nous ajoutons à ce niveau les modes de présentation innovants, mis au point à l'aune du profil des développeurs logiciels : arbres de décision, types d'applications et aperçus de patrons.

L'ensemble des patrons et des modes de présentation forme la bibliothèque de patrons de CAE. Celle-ci est partagée de manière restreinte sur le réseau intranet de l'entreprise via un système de gestion de contenu en source libre, ce qui nous permet de mener quelques séances d'essai semi-formelles pour la valider.

Les modes de présentations ont alors été comparés au mode de présentation de référence tel qu'identifié dans la revue de littérature (un index alphabétique simple) dans des tâches de recherche de patron. Nous avons pris soin de sélectionner des participants n'ayant jamais utilisé de patrons de conception d'IHM ni vu la bibliothèque au préalable. Nos résultats montrent que si les participants trouvent des patrons 37 fois sur 40 tentatives, les temps de recherche de patrons augmentent de 51% avec les types d'application, de 23% avec les arbres de décision et de 11% avec les aperçus de patrons par rapport au mode de référence. La satisfaction des utilisateurs augmente de 42% avec les types d'applications et les arbres de décision et de 73% avec les aperçus de patrons. En outre, trois participants sur 10 disent préférer les types d'applications, sept sur 10 les aperçus et aucun ne se prononce pour les arbres de décision ou l'index de référence. Par ailleurs, les patrons trouvés avec les types d'application, les arbres de décision et les aperçus de patrons sont respectivement 89%, 100% et 50% plus pertinents que ceux obtenus avec l'index alphabétique.

Ces résultats nous conduisent à recommander l'usage de modes de présentation alternatifs des patrons (types d'applications, arbres de décision et aperçus) pour une bibliothèque destinée en partie ou en totalité à des développeurs logiciels. Le mode de présentation de référence (index alphabétique) devrait cependant être conservé car sa simplicité rassure les développeurs logiciels.

L'amplitude des gains donne à penser que les modes de présentations pourraient être une composante fondamentale des bibliothèques ou des langages de patrons, jusqu'ici négligés à cause de leur nature extrinsèque aux patrons.

Enfin, les patrons eux-mêmes devraient être écrits dans un style simple et clair faisant usage de listes à points et d'une grande quantité d'exemples soigneusement choisis.

Pistes de recherche & travaux futurs

Il conviendrait de vérifier les résultats précédents par une étude de plus grande ampleur, impliquant plus de participants, avec des profils professionnels plus variés et surtout s'inscrivant

dans la durée. En effet, notre expérimentation ne concerne que le premier contact des utilisateurs avec la bibliothèque. Il serait intéressant d'observer le comportement des utilisateurs dans la durée et de modifier, ajouter ou retirer des modes de présentation en fonction de leur apprentissage et de leur appropriation de la bibliothèque.

De même, il faudrait analyser de manière plus formelle l'influence effective des patrons de conception sur la qualité des IHM produites. La discipline manque d'une méthode standard permettant de mener une telle analyse et donc de comparer la performance de patrons, de bibliothèques ou de modes de présentation.

La bibliothèque de patrons de CAE est encore en développement et recèle de pistes à explorer, depuis la formalisation de l'évangélisation d'une bibliothèque de patrons jusqu'à la mise au point d'autres modes de présentation, toujours dans un objectif d'accessibilité aux développeurs logiciels.

Finalement, la considération des modes de présentation comme une fonction à part entière d'une bibliothèque ouvre la voie à une approche différente des patrons de conception. En s'écartant des acquis implicites qui régissent la plupart des bibliothèques actuelles, nous pourrions repenser la nature et l'importance des facteurs qui devraient ou pourraient composer une bibliothèque de patron et son contenu.

BIBLIOGRAPHIE

Alexander, C. (1964). *Notes on the Synthesis of Form*: Harvard University Press.

Alexander, C. (1979). *The Timeless Way of Building*: Oxford University Press.

Alexander, C., Ishikawa, S. & Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction*: Oxford University Press.

BBC. (2010). BBC - GEL (Global Experience Language) - Design Patterns Récupéré le 03/04/2011 de http://www.bbc.co.uk/guidelines/gel/dp_localnav.shtml

Beck, K., Crocker, R., Meszaros, G., Vlissides, J., Coplien, J. O., Dominick, L. et Paulisch F. (1996). *Industrial experience with design patterns*. Proceedings of the 18th international conference on Software engineering. IEEE Computer Society.

Behrens, C. (2008). InfoDesignPatterns.com. Récupéré le 03/03/2011 de <http://www.niceone.org/infodesignpatterns/index.php5#/patterns.php5>

Berriman, F., Cederholm, D., Çelik, T., Khare, R., King, R., Marks, K. et Ward, B. (2011). Microformats | About Microformats. Récupéré le 03/03/2011 de <http://microformats.org/about>

Blackwell, A. F., & Fincher, S. (2010). PUX: patterns of user experience. *interactions*, 17(2), 27-31.

Booch, G. (2003). *Core J2EE Patterns: Best Practices and Design Strategies (Foreword)* (2nd ed.): Prentice Hall.

Breiner, K., Seissler, M., Meixner, G., Forbrig, P., Seffah, A. et Klöckner, K. (2010). *PEICS: towards HCI patterns into engineering of interactive systems*. Proceeding of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems. ACM.

CAE. (2010). CAE › À propos de CAE › Foire aux questions. Récupéré le 03/03/2011 de <http://www.cae.com/fr/about.cae/faqs.asp>

CAE. (2010). Un chef de file mondial (Introduction à CAE). Matériel promotionnel interne, Montréal.

Constantine, L. & Lockwood, L. (2003). forUse: The Electronic Newsletter of Usage-Centered Design #29 | February 2003. *forUse: The Electronic Newsletter of Usage-Centered Design*. Récupéré le 03/03/2011 de <http://www.foruse.com/newsletter/foruse29.htm#2>

Cooper, A. & Reimann, R. (2003). *About Face 2.0: The Essentials of Interaction Design* (2nd ed.): Wiley.

Coram, T. & Lee, J. (1998). Experiences -- A Pattern Language for User Interface Design. Récupéré le 04/04/2011 de <http://www.maplefish.com/todd/papers/experiences/Experiences.html>

Crumlish, C. & Malone, E. (2009). *Designing Social Interfaces: Principles, Patterns, and Practices for Improving the User Experience*: Yahoo Press.

Cunningham, W. & Beck, K. (1987). *Using Pattern Languages for Object-Oriented Programs*. OOPSLA-87 workshop on the Specification and Design for Object-Oriented Programming.

Curtis, N. (2008). Pattern Library vs. Component Library: What's the Diff? Récupéré le 03/03/2011 de <http://www.nathancurtis.com/2008/02/21/pattern-library-vs-component-library-whats-the-diff/>

Curtis, N. (2009). Components Versus Patterns. Récupéré le 03/03/2011 de http://www.uie.com/articles/components_vs_patterns/

Davenport, T. H. (1995). Some Principles of Knowledge Management. *Business and Strategy*, 34-41.

Dearden, A. & Finlay, J. (2006). Pattern Languages in HCI: A Critical Review. *Human-Computer Interaction*, 21(1), 49 - 102.

Duyne, D. K. v., Hong, J. I. & Landay, J. A. (2007). *Design of Sites: Patterns for Creating Winning Websites*: Prentice Hall.

Erickson, T. (2000). *Lingua Francas for design: sacred places and pattern languages*. Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques.

Fincher, S. & Utting, I. (2002). *Pedagogical patterns: their place in the genre*. Proceedings of the 7th annual conference on Innovation and technology in computer science education. ACM.

- Fincher, S. & Windsor, P. (2000). *Why patterns are not enough: some suggestions concerning an organising principle for patterns of UI design*. CHI'2000 Workshop on Pattern Languages for Interaction Design: Building Momentum.
- Gaffar, A. (2005). *Studies on pattern dissemination and reuse to support interaction design*. Concordia University, Montréal.
- Gaffar, A., Seffah, A. & Poll, J. A. V. d. (2005). *HCI pattern semantics in XML: a pragmatic approach*. Proceedings of the 2005 workshop on Human and social factors of software engineering. ACM.
- Gaffar, A., Sinnig, D., Javahery, H. & Seffah, A. (2003). *MOUDIL: A Comprehensive Framework for Disseminating and Sharing HCI Patterns*. CHI 2003 workshop on HCI Patterns: Concepts and Tools.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software* (1 ed.): Addison-Wesley Professional.
- Granlund, A., Lafrenière, D. & Carr, D. A. (2001). *A Pattern-Supported Approach to the User Interface Design Process*. Proceedings of HCI International 2001 (9th International Conference on Human-Computer Interaction), New Orleans, USA.
- Grover, V. & Davenport, T. H. (2001). General Perspectives on Knowledge Management: Fostering a Research Agenda. *J. Manage. Inf. Syst.*, 18(1), 5-21.
- Gruber, J. (2004). Daring Fireball: Markdown. Récupéré le 03/04/2011 de <http://daringfireball.net/projects/markdown/>
- Hamari, J., & Eranti, V. (2011). *Framework for designing and evaluating game achievements*. Helsinki Institute for Information Technology HIIT. Récupéré le 03/04/2011 de <http://www.slideshare.net/JuhoHamari/framework-for-designing-and-evaluating-game-achievements>
- Hickson, I. (2011). *HTML Microdata*: W3C.
- Janeiro, J., Barbosa, S., Springer, T. & Schill, A. (2010). Improving the Search for User Interface Design Patterns through Typed Relationships. In P. Forbrig, F. Paternó & A. Mark Pejtersen (Eds.), *Human-Computer Interaction* (Vol. 332, pp. 3-14): Springer Boston.

- Janeiro, J., Barbosa, S. D. J., Springer, T. & Schill, A. (2010). *Semantically relating user interface design patterns*. Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems. ACM.
- Javahery, H. (2007). *Pattern-oriented UI design based on user experiences: a method supported by empirical evidence*. Concordia University, Montreal.
- Javahery, H. & Seffah, A. (2002). *A Model for Usability Pattern-Oriented Design*. Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design.
- Kawasaki, G. (2004). *The Art of the Start: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything* Portfolio Hardcover.
- Krug, S. (2005). *Don't Make Me Think: A Common Sense Approach to Web Usability* (2nd ed.): New Riders Press.
- Kruschitz, C. & Hitz, M. (2009). *The Anatomy of HCI Design Patterns*. Proceedings of the 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns.
- Kruschitz, C. & Hitz, M. (2010). *Are human-computer interaction design patterns really used?* Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries.
- Kruschitz, C. & Hitz, M. (2010). *Bringing formalism and unification to human-computer interaction design patterns*. Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems.
- Lammi, J. (2007). *Developing a UI Design Pattern Library - A Case Study at eCraft*. Helsinki University of Technology.
- Leacock, M., Malone, E. & Wheeler, C. (2005). *Implementing a Pattern Library in the Real World: A Yahoo! Case Study*. ASIS&T IA Summit.
- Leacock, M., Malone, E. Wheeler, C., & Scott, B. (2006). Yahoo! Design Pattern Library. Récupéré le 03/03/2011 de <http://developer.yahoo.com/ypatterns/>
- Lejeune, M. (2008). SSH 5100 Sociologie de la technologie: École Polytechnique de Montréal.

Limpens, F., Gandon, F. & Buffa, M. (2009). *Linking Folksonomies and Ontologies for Supporting Knowledge Sharing: a State of the Art*.

Magnuson, E. (2010). *Design Patterns in User Interface Design*. Worcester Polytechnic Institute.

Mahemoff, M. & Johnston, L. (1998). *Pattern Languages for Usability: An Investigation of Alternative Approaches*. Proceedings of the Third Asian Pacific Computer and Human Interaction.

Mahemoff, M. & Johnston, L. J. (2001). *Usability Pattern Languages: the "Language" Aspect*. Human-computer interaction: INTERACT '01.

Mahemoff, M. J. & Johnston, L. J. (1998). *Principles for a Usability-Oriented Pattern Language*. Proceedings of the Australasian Conference on Computer Human Interaction.

Malone, E. (2010). Working With a Pattern Library Day to Day. Récupéré le 03/03/2011 de <http://www.emdezine.com/deziningInteractions/2010/07/29/working-with-a-pattern-library-day-to-day/>

Malone, E. (2010). How to write an interaction design pattern. Récupéré le 15/02/2011 de <http://www.emdezine.com/deziningInteractions/2010/04/20/how-to-write-an-interaction-design-pattern/>

Mayhew, D. J. (1999). *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design (Interactive Technologies)*: Morgan Kaufmann.

Meszaros, G. & Doble, J. (1997). A pattern language for pattern writing *Pattern languages of program design 3* (pp. 529-574): Addison-Wesley Longman Publishing Co., Inc.

Migneault, J. (2008). *Étude de l'intégration de la reconnaissance vocale au poste d'opération de l'instructeur en simulateur de vol*. École Polytechnique de Montréal, Montréal.

Mika, P., Atserias, J., Ciaramita, M. & Zaragoz, H. (2008). *Exploiting explicit and implicit semantics on the Web*. SIKS Seminar on the Semantic Web, Utrecht.

Morville, P. & Callender, J. (2010). *Search Patterns: Design for Discovery*: O'Reilly Media.

Myers, B. A. & Rosson, M. B. (1991). User interface programming survey. *SIGCHI Bulletin*, 23(2), 27-30.

- Neil, T. (2010). 12 Standard Screen Patterns. Récupéré le 03/02/2011 de <http://designingwebinterfaces.com/designing-web-interfaces-12-screen-patterns>
- Nesladek, C., Bauer, G., Fulcher, R., Robertson, C. & Palmer, J. (2010). Android UI design patterns: Google I/O 2010.
- Nielsen, J. (2005). Ten Usability Heuristics. Récupéré le 10/04/2011 de http://www.useit.com/papers/heuristic/heuristic_list.html
- ISO (1998). Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) (Vol. Partie 11: Lignes directrices relatives à l'utilisabilité): ISO.
- ISO (2010). Ergonomie de l'interaction homme-système (Vol. Partie 210: Conception centrée sur l'opérateur humain pour les systèmes interactifs): ISO.
- Norman, D. (2002). *The Design of Everyday Things*. New York: Basic Books.
- Reffell, J. (2006). Design Pattern Conversation: What's the Best Way to Communicate Patterns? Part Five. Récupéré le 03/03/2011 de http://www.yuiblog.com/blog/2006/10/23/communicating_patterns_part_five/
- Robert, J. M. (2009). Les procédures de travail, *IND6406 Ergonomie cognitive*: École Polytechnique de Montréal.
- Salinas, N. A. (2000). The Structure of Pattern Languages. *Architectural Research Quarterly*, 4, 149-161.
- Saunders, W. S. (2002). A Pattern Language. *Harvard Design Magazine*.
- Scott, B. (2006). Design Patterns: Part 3 - What do we mean by "design patterns"? *Design Patterns*. Récupéré le 03/03/2011 de <http://www.lukew.com/ff/entry.asp?350>
- Seffah, A. (2010). *The evolution of design patterns in HCI: from pattern languages to pattern-oriented design*. Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems.
- Seffah, A. & Javahery, H. (2002). *On the Usability of Usability Patterns*. Workshop Patterns in Practice, CHI.

Sinnig, D. (2004). *Enhancing Reuse: Towards Pattern- and Component- Based UI Development*. University of Rostock.

Sinnig, D., Javahery, H., Forbrig, P., & Seffah, A. (2005). *Patterns and Components for Enhancing Reusability and Systematic UI Development*. Human-Computer Interaction.

Spool, J. (2009). UIEtips: Components, Patterns, and Frameworks! Oh My! *UIEtips* Récupéré le 03/02/2011 de <http://www.uie.com/brainsparks/2009/05/20/uietips-componentspatternsframeworks/>

Spool, J. (2009). Components, Patterns, and Frameworks! Oh My! Récupéré le 03/02/2011 de <http://www.uie.com/articles/componentspatternsframeworks/>

Stapleton, P. (2011). UI Pattern Documentation Review. Récupéré le 03/03/2011 de <http://www.bboxesandarrows.com/view/ui-pattern>

Taleb, M., Seffah, A. & Abran, A. (2007, 13-15 Aug. 2007). *Patterns-Oriented Design Applied to Cross-Platform Web-based Interactive Systems*. Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on.

Tidwell, J. (1997). COMMON GROUND: A Pattern Language for Human-Computer Interface Design. Récupéré le 03/03/2011 de http://www.mit.edu/~jtidwell/common_ground.html

Tidwell, J. (2005). *Designing Interfaces: Patterns for Effective Interaction Design*. {O'Reilly Media, Inc.}.

Tidwell, J. (2006). Design Pattern Conversation: What's the Best Way to Communicate Patterns? Part One. *Design Pattern Conversation: What's the Best Way to Communicate Patterns?* Récupéré le 03/03/2011 de http://www.yuiblog.com/blog/2006/10/11/communicating_patterns_part_one/

Tidwell, J. (2007). *Designing "Designing Interfaces:" How Not to Write a Pattern Catalog*. Article présenté à UPA 2007.

UNITiD. (2010). Android Interaction Design Patterns. Récupéré le 03/03/2011 de <http://www.androidpatterns.com/>

Vlissides, J. (1997). Patterns: the top ten misconceptions. *Object Magazine*.

Watters, A. (2010). 10 Tips for Adding Game Mechanics to a Non-Gaming Service. Récupéré le 03/02/2011 de <http://www.readwriteweb.com/start/2010/09/10-tips-for-adding-game-mechan.php>

Welie, M. v. (2000). A Pattern Library for Interaction Design. Récupéré le 03/03/2011 de <http://www.welie.com/>

Welie, M. v. (2006). Design Pattern Conversation: What's the Best Way to Communicate Patterns? Part Four. *Design Pattern Conversation: What's the Best Way to Communicate Patterns?* Récupéré le 03/03/2011 de http://www.yuiblog.com/blog/2006/10/20/communicating_patterns_part_four/

Welie, M. v. & van der Veer, G. (2003). *Pattern Languages in Interaction Design: Structure and Organization*. Proceedings of Interact '03, Zurich, Switzerland.

Wieringa, D., Moore, C. & Barnes, V. (1999). *Procedure Writing: Principles and Practices*. Institute of Electrical & Electronics Engineer.

ANNEXE A – Revue de littérature

Cette annexe présente et compare les bibliothèques de patrons qui forment notre revue de littérature. Leurs caractéristiques générales sont d’abord examinées dans trois tableaux comparatifs :

- Le premier introduit les informations générales des bibliothèques,
- Le deuxième présente le public visé et les sujets couverts.

Si le *public visé* n’est pas explicitement mentionné par l’auteur, alors on le remplace par le cas d’usage recommandé par l’auteur.

La colonne *Relations entre patrons* décrit la manière dont les patrons d’une même bibliothèque sont reliés entre eux.

Les *pointeurs* sont l’équivalent papier des liens hypertextes.

- Le troisième décrit les modes de présentation des patrons dans la bibliothèque. C’est cet aspect qui nous intéresse le plus dans le cadre du mémoire.

Les schémas de patrons de chaque bibliothèque sont présentés dans les visualisations de l’annexe B. Enfin, sauf mention contraire, les patrons sont présentés dans les bibliothèques sans commentaires ou explication.

A.1 Présentation des bibliothèques revues

Nom	URL	Forme	Nombre de patrons
Designing Interfaces <i>Patterns for Effective Interaction Design</i>	http://designinginterfaces.com/	Livre	125
Designing Web Interfaces <i>Principles and Patterns for Rich Interaction</i>	http://designingwebinterfaces.com/	Livre	65
Search Patterns <i>Design for Discovery</i>	http://searchpatterns.org/	Livre	10
The design of sites <i>Patterns for creating winning web sites</i>	http://www.designofsites.com/home/	Livre	107
Design Patterns Library <i>Fellowship Technologies</i>	http://developer.fellowshipone.com/patterns/	Site Web	25
Experiences <i>A Pattern Language for User Interface Design</i>	http://www.maplefish.com/todd/papers/experiences/Experiences.html	Site Web	26
UI Design Pattern Library <i>Patternry</i>	http://www.patternry.com/	Site Web	32
UI-Patterns.com	http://ui-patterns.com/	Site Web	56
User Interface Design Patterns	http://www.cs.helsinki.fi/u/salaakso/patterns/	Site Web	21
Welie.com <i>Patterns in Interaction Design</i>	http://www.welie.com/index.php	Site Web	131
Yahoo! Design Pattern Library	http://developer.yahoo.com/ypatterns/	Site Web	59

A.2 Public visé, thèmes couverts et relations entre patrons

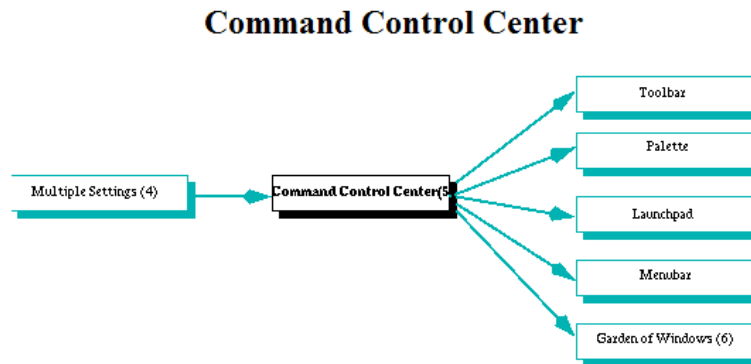
Nom	Public visé (traduction libre)	Thèmes couverts	Relations entre patrons
Designing Interfaces <i>Patterns for Effective Interaction Design</i>	« Développeurs logiciels, concepteurs de pages Web, nouveaux concepteurs d'interface et spécialistes de l'utilisabilité, concepteurs plus expérimentés (...). Une connaissance de base de la conception d'IHM est requise. »	Ordinateurs, PDA, sites Web, applications Web, téléphones intelligents, téléphones <i>(présumé)</i>	Pointeurs dans le corps de texte
Designing Web Interfaces <i>Principles and Patterns for Rich Interaction</i>	« Quiconque spécifiant, concevant ou développant des interfaces Web (...) Concepteurs Web, gestionnaires de produit, développeurs Web. »	Sites Web, applications Web	Pointeurs dans le corps de texte
Search Patterns <i>Design for Discovery</i>	«Designers, information architects, student, entrepreneurs and anyone who cares about the future of search»	Ordinateurs, sites Web, applications Web, téléphones intelligents	Aucun
The design of sites <i>Patterns for creating winning web sites</i>	« Concepteurs, professionnels de la conception Web »	Sites Web, applications Web	Pointeurs dans le corps de texte
Design Patterns Library <i>Fellowship Technologies</i>	« pour les applications Web (...) telles que spécifiées par l'équipe d'Expérience Utilisateur »	Applications Web	Aucun
Experiences <i>A Pattern Language for User Interface Design</i>	Non précisé	Ordinateurs <i>(présumé)</i>	Liens hypertextes dans le corps de texte
UI Design Pattern Library <i>Patternry</i>	« Quiconque voulant créer des applications ou sites Web formidables »	Sites Web, applications Web	Liens hypertextes dans le corps de texte
UI-Patterns.com	« Concepteur chevronné d'interfaces utilisateur »	Sites Web	Liens hypertextes dans une section à part
User Interface Design Patterns	« Étudiants des cours de conception de l'Université d'Helsinki »	Ordinateurs, PDA, téléphones intelligents	Liens hypertextes dans le corps de texte
Welie.com <i>Patterns in Interaction Design</i>	« ...lors de la conception d'expériences utilisateur »	Sites Web	Liens hypertextes dans le corps de texte
Yahoo! Design Pattern Library	« Communauté de la conception et du développement Web »	Sites Web, applications Web	Aucun

A.3 Modes de présentation & lisibilité machine

Nom	Mode de présentation	Lisibilité machine
Designing Interfaces <i>Patterns for Effective Interaction Design</i>	Liste découpée en groupes d'une quinzaine de patrons. Chaque groupe est introduit par une dizaine de pages de commentaires. Certains pointeurs sont commentés.	Non
Designing Web Interfaces <i>Principles and Patterns for Rich Interaction</i>	Groupes de deux à neuf patrons. Ces groupes sont organisés en 6 sections. Chaque groupe et chaque section est introduit par une page de commentaire. Tous les pointeurs sont commentés.	Non
Search Patterns <i>Design for Discovery</i>	Aucun. Les patrons ne sont pas énumérés, ils sont directement imprimés les uns à la suite des autres.	Non
The design of sites <i>Patterns for creating winning web sites</i>	Liste découpée en 13 groupes de deux à 17 patrons. Chaque groupe et chaque patron sont décrits par 3 lignes de texte.	Non
Design Patterns Library <i>Fellowship Technologies</i>	Liste découpée en groupe de trois à sept patrons. Deux patrons sont divisés en deux sous-patrons.	Non
Experiences <i>A Pattern Language for User Interface Design</i>	Liste découpée en groupe de un à 10 patrons. Les groupes ont une phrase de commentaire. Une visualisation de la bibliothèques est proposée.	Non
UI Design Pattern Library <i>Patternry</i>	Liste tous les patrons d'un coup (titre + capture d'écran d'exemple)	Non
UI-Patterns.com	Liste découpée en groupes de un à 12 patrons. Les groupes sont	Non
User Interface Design Patterns	Liste découpée en groupes de deux à quatre patrons.	Non
Welie.com <i>Patterns in Interaction Design</i>	Liste découpée en groupes de trois à 25 patrons. Ces groupes sont organisés en 3 sections, chacune recevant une phrase de commentaire.	Oui (PLML)
Yahoo! Design Pattern Library	Deux méthodes sont offertes en parallèle : - Une liste plate et riche (titre + exemple + solution) de tous les patrons, éventuellement divisée en 5 groupes définis par un titre et une phrase, - Une table des matières de liens simples, divisées en 5 groupes avec encore un niveau de subdivision possible.	Non

A.4 Exemples de patrons

Command Control Center (Experiences)



...when your application has [Multiple Settings](#), each application must be easily accessed and integrated in order to create the sense that everything is leading the user to reaching the established goal.

Users need a well defined mechanism to initiate tasks and navigate among them.

Users want a consistent interface that is always present and easy to reach, an interface they can use to instruct the application to do their bidding.

When there are many tools, windows, or areas of concentration, users will need a mechanism to get there. And without a central navigation center, users will be forced to search for previously opened applications. This may result in two or more instances of the same application because the user could not find the previous instance.

Separate windows in an application should not exist without any well defined way to control them. This is especially true in a multitasking environment, where multiple distinct applications may be running at the same time. The user needs to know what visible or iconified window belongs to a particular application.

Consider an integrated development environment (IDE) that allows the users to bring up the debugger only from the compiler window's menu bar. What if the user wishes to debug a previously compiled program? How will she know that she needs to bring up the compiler window to access the debugger? An intuitive solution is needed.

Therefore:

Provide a command control center that is always present, easily accessible and provides quick access to tools, tasks and windows. If a single control center window cannot be provided, present your control center as commands that are accessible from every window's menubar.

The command control center acts as the "home base" for coordinating applications within Multiple Settings, or for coordinating secondary windows within a Garden of Windows.

In Symantec Cafe (a Java program development tool), users know they need to perform tasks such as edit, compile, run and debug. But, how do they navigate through these tasks? They use menubars, toolbars and other controls to reach each task. Once these tools are identified, they should be easy to locate and provide navigation among tasks as seamlessly as possible. In Cafe, each task is associated with a window. A command center model is adopted that makes it easy to select tasks by choosing windows.

The Windows 95 *Start Bar* is always visible and always contains the names of the running applications. One simple click of the mouse on a name brings any application to the foreground. The users remember where these controls are and rely on them to *get around*.

When building your control center, you may use a [Menubar](#), or perhaps a [Launchpad](#) to provide navigation among your [Garden of Windows](#), while a [Toolbar](#) or [Palette](#) to provide access to tools...

Home Link (Welie.com), commentaires non montrés.

Home Link

Problem

Users need to get back to a safe/start/familiar point

Solution

Use a fixed element, such as the site's logo, as a link to the home page



Use when

In most Web/WAP sites there is a home page. This page is usually the starting point for interaction. When users arrive at an arbitrary page within the site, they should be able to go back to the home page.

How

Add a link to the home-page on every page. Place the link at the top of the page and if appropriate on the bottom of the screen. If the site has a logo, make the logo a link to the home-page as well. Otherwise, simply link the home-page using the label "home", either a text label or an icon of a house. Make sure the link is always found on the same place.

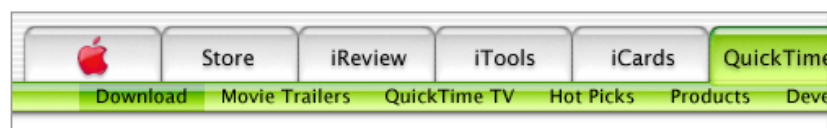
If the site has sub-sites, you need to make a distinction between the root home and the local home. In that case make the logo the local home and provide a link to the root home.

Why

A home link provides a safe exit on every page and always on the same position. No matter what, the users can always get back to a familiar place. Logo's identify the site and are therefore easily chosen to get back to the home page.

More Examples

This example of the Apple website shows how the Apple logo is used as a link to the home page.



A.5 Visualisation des modes de présentation

Nous avons modélisé quelques bibliothèques sous forme graphique. Il s'agit d'une visualisation des *modes de présentation*, c'est-à-dire de la structure hiérarchique des patrons tel que proposé à l'utilisateur, et non de tous les liens qui existent entre patrons : les liens horizontaux, d'un patron vers un autre, n'y figurent pas.

Les libellés en gras sont les titres des groupes ou des sections de la bibliothèque ; les libellés en texte simple ses patrons.

Designing Interfaces

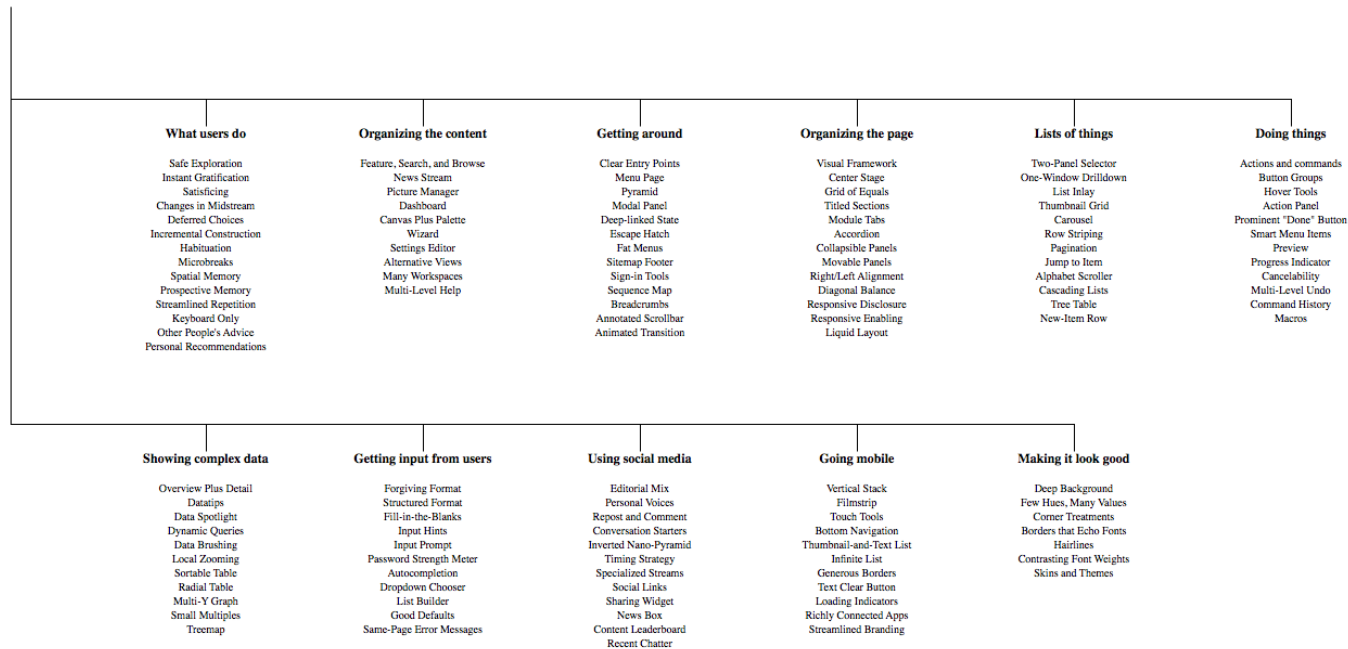


Schéma de patron :

- Titre
- Exemple sensibilisant
- Quoi
- Utiliser quand
- Pourquoi
- Comment
- Exemples
- Dans d'autres bibliothèques (patrons similaires)

Designing Web Interfaces

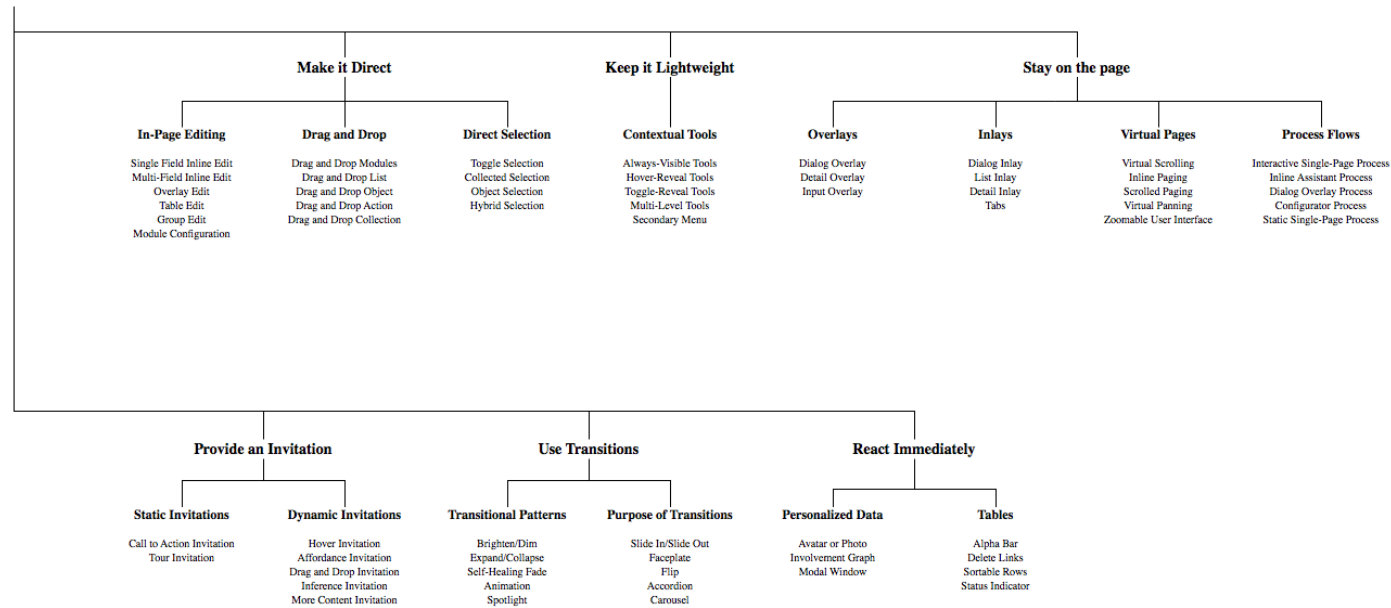


Schéma de patron :

- Titre
- Description
- Exemples
- Considérations

The design of sites

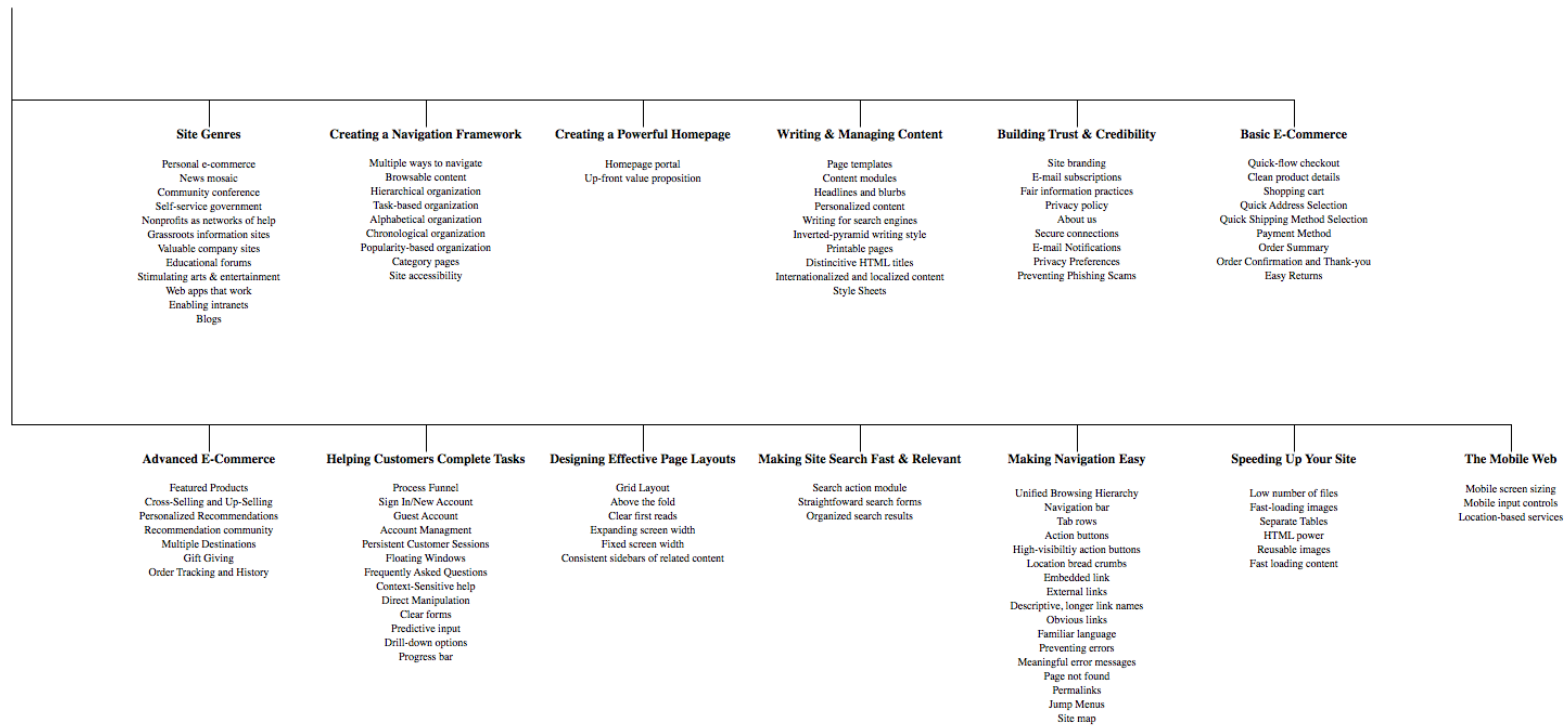


Schéma de patron :

- Titre
- Exemple sensibilisant
- Contexte
- Problème
- Solution
- Autres patrons à considérer

Design Patterns Library

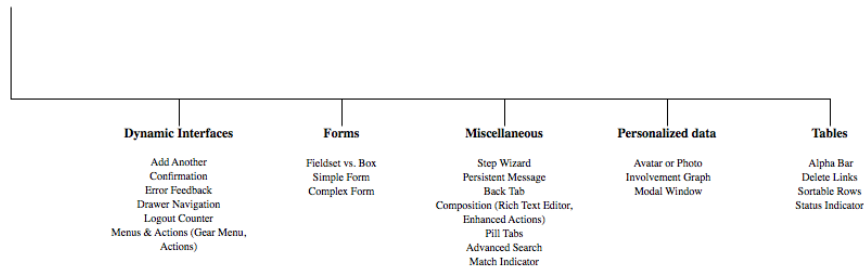


Schéma de patron :

- Titre
- Visuel d'exemple (réalisé exclusivement avec les outils internes)
- Problème
- Patron
- Résultat
- (Code)

Welie.com

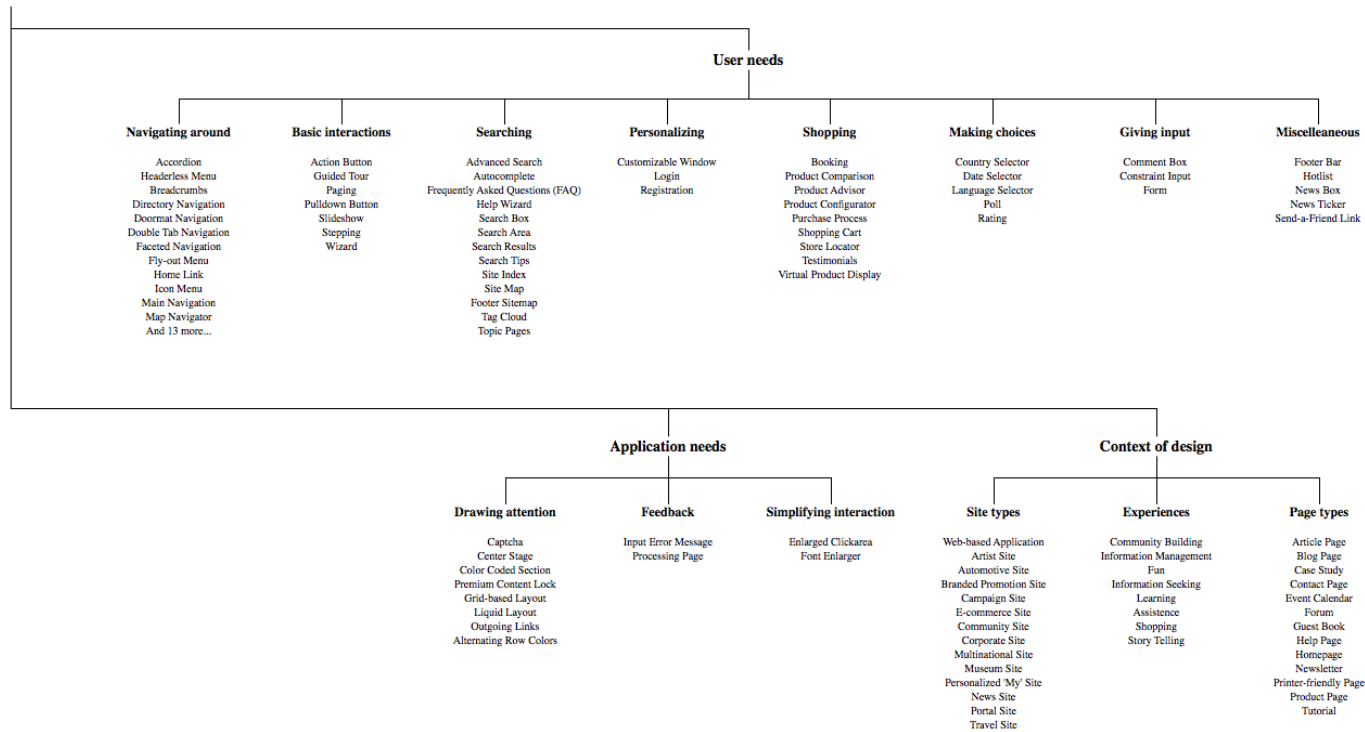


Schéma de patron :

- Titre
- Problème
- Solution (avec exemple)
- Utiliser quand
- Comment
- Pourquoi
- Plus d'exemples
- Commentaires (ouverts au public)

ANNEXE B – Système de gestion de contenu

B.1 Solutions comparées

Système	Avantages	Inconvénient	Commentaire
Drupal <i>Système de gestion de contenu</i>	Système de taxonomie unique permettant un excellent support du principe structurant d'une bibliothèque.	Techniquement difficile à mettre en œuvre, Communauté d'utilisateurs restreinte.	Trop exigeant pour nos ressources techniques.
Expression Engine <i>Système de gestion de contenu</i>	Bon ensemble de fonctionnalités, support commercial.	Communauté d'utilisateurs restreinte. Payant.	Trop exigeant pour nos ressources techniques.
FireRift <i>Système de gestion de contenu</i>	Facilité de mise en œuvre et de publication, excellent support multimédia, souplesse du système.	Peu de fonctionnalités, Communauté d'utilisateurs très restreinte.	Doutes sur la pérennité du produit.
Sharepoint <i>Gestion électronique de documents</i>	Déjà utilisé par CAE (familier pour les développeurs logiciels).	Pas de portabilité des données, Utilisabilité faible, difficulté de publication, présentation fixe.	Destiné à gérer des documents plutôt que des connaissances.
Stacey <i>Système de gestion de contenu</i>	Facile à modifier, Léger en ressources.	Fonctionnalités limitées (pas d'étiquette par exemple), Pas s'éditeur de texte graphique.	Plus adapté pour de petites équipes, à l'aise techniquement.
Textpattern <i>Moteur de blogue</i>	Facile d'utilisation, Excellent séparation des différents rôles d'utilisateurs (ergonomes en écriture, développeurs logiciels en lecture).	Communauté d'utilisateurs restreinte.	Supporte Markdown. Décline au profit de produits plus populaires (notamment Wordpress).
TiddlyWiki <i>Moteur de wiki</i>	Facile d'utilisation, Léger en ressources.	Fragile.	Développé partiellement par Michael Mahemoff, un expert des patrons
Wikimedia <i>Moteur de wiki</i>	Bonne adéquation avec le besoin de la bibliothèque.	Utilisabilité faible, difficulté de publication, mauvais support multimedia.	Théoriquement proche de nos besoins, mais peu pertinent dans les faits : top compliqué, trop lent...
Wordpress <i>Moteur de blogue</i>	Immense support de la communauté (disponibilité de nombreux modules de fonctionnalités supplémentaires).	Code et interface parfois brouillons.	Supporte Markdown. Système finalement choisi pour la bibliothèque de CAE.

Toutes les solutions examinées sont gratuites, à l'exception de Expression Engine et Sharepoint.

La communauté d'utilisateurs est un facteur important à prendre en compte car elle est susceptible de fournir du support ou des modules améliorant la fonctionnalité ou l'apparence des systèmes présentés.

Les systèmes de gestion de contenu hébergés, c'est-à-dire exécutés depuis un prestataire de service plutôt que sur un serveur interne à CAE, n'ont pas été examinés : leur emploi est interdit pour des raisons de sécurité. Cela concerne notamment Quince (<http://quince.infragistics.com/>) et Patternry (<http://patternry.com/>).

La bibliothèque de patrons de CAE a d'abord été hébergée dans un système de gestion de contenu rudimentaire fait maison. Lorsqu'il a fallu monter en puissance (plus de contenu, d'utilisateurs, de fonctions...), cette approche exigeante en ressources techniques a été abandonnée.

B.2 Configuration de la solution choisie

1. Installation d'un serveur WAMP sur l'intranet de CAE.
2. Installation de Wordpress 3.0.1 avec le thème Thematic 0.9.7.7
3. Installation du module « Markdown for WordPress and bbPress », désactivation de l'éditeur visuel de Wordpress
4. Installation du module « Search Unleashed »
5. Création d'un sous-thème « Pieuvre » :
 - Nouvelle apparence visuelle
 - Codage d'une fonction PHP retournant les aperçus de patrons plutôt qu'une chaîne de caractères lorsque l'utilisateur fait une recherche ou un tri des entrées de la bibliothèque.
 - Codage d'une fonction PHP permettant de mettre des arbres de décision sous forme d'assistant dans le site via le dévoilement animé et progressif des branches.

ANNEXE C – Bibliothèque de patrons CAE

C.1 Liste des patrons CAE

ID	Nom	Spécificité	Longueur	Terminé
29	Écran d'accueil	3	201	Oui
33	Envers du décor	3	152	Oui
211	Pile de notifications	2	35	Non
213	Aperçu	1	30	Non
216	Paquetage	1	241	Oui
218	Issue de secours	1	27	Non
220	Le complexe possible	1	55	Non
222	Témoin de progression	1	181	Oui
226	Indices ambiants	1	141	Non
228	Bonne typographie	1	30	Non
230	Hiérarchie claire	1	30	Non
232	Étagère temporaire	2	26	Non
234	Accélérateurs	2	32	Non
236	Objet Riche	1	58	Oui
238	Interaction dans le contenu	2	108	Oui
240	Zoom sémantique	3	44	Non
242	Manipulation directe	2	29	Non
244	Commandes contextuelles	2	33	Non
246	Transition plein-écran	3	151	Oui
248	En voir plus	1	30	Non
250	Mega menu déroulant	2	181	Oui
252	Gerty	2	44	Non
254	Cape des ombres	3	31	Non
256	Panorama	2	68	Non
258	Hovercard	2	44	Non
260	Bonne couleur	1	35	Non
262	Échec et mat	2	31	Non

La *longueur* des patrons correspond à leur nombre de lignes dans l'éditeur de code qui sert à les rédiger.

Les patrons ne sont pas tous *terminés*, c'est-à-dire prêts pour la production. Certains n'ont que leur aperçu et quelques lignes de texte, d'autres sont quasiment terminés.

C.2 Types d'application CAE

« Ils recensent les “grandes familles” de logiciels pour isoler l'ensemble de patrons adapté à votre cas. »

Nom	Description
Système de gestion de contenu	Depuis les CMS dédiés comme SharePoint jusqu'aux panneaux de gestion des cas de Owl.
Modèle immersif	On agit sur une représentation graphique d'une carte, d'un avion, d'un patient...
Multimedia	Pour les écrans dédié à la gestion ou à la lecture de photos, de vidéos, de musique...
Préférences	“Réglages”, “Options”... pouvant s'appliquer à tout le logiciel ou juste à un de ses composants.

C.3 Arbres de décision CAE

« Ils permettent de trouver le bon patron pour votre problème en répondant à quelques questions successives. »

Nom	Description
Remplacer des fenêtres enfants	Les fenêtres enfant sont pénibles pour l'utilisateur ; trouvez ici comment les remplacer au mieux.
Améliorer la qualité perçue	Astuces pour embellir un UI à moindre frais.
Limiter le taux d'erreur	Éviter perte de temps et de données à l'utilisateur, il vous remerciera en ne jetant pas sa machine par la fenêtre.
Fournir du feedback à l'utilisateur	Savoir ce qui s'est passé, ce qui se passe et ce qui va se passer est crucial pour contrôler un système.

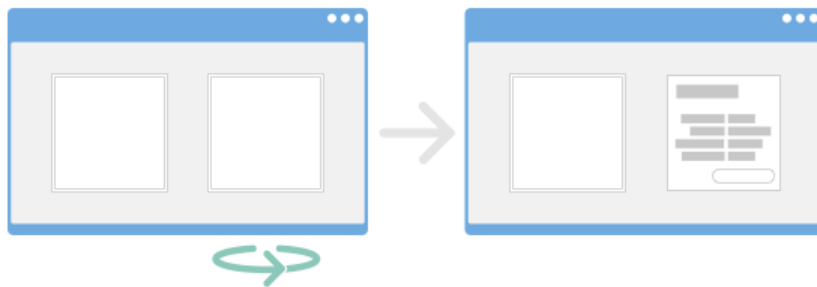
C.4 Exemple de patron CAE

Envers du décor, avec section *Exemples* tronquée au cinquième de sa taille originale pour des raisons d'encombrement.

Envers du décor

By JOCELYN | Published: NOVEMBER 22, 2010 | Edit

Afficher tous les paramètres « au verso » du logiciel.



Problème

L'utilisateur veut voir ou modifier les paramètres d'un logiciel.

Contexte

Les paramètres de l'application sont nombreux ; ils nécessitent beaucoup d'espace ou plusieurs fenêtres pour être affichés. On peut aussi avoir un nombre raisonnable de paramètres mais ne pas pouvoir les afficher dans une boîte de dialogue, par exemple parce que la plate-forme utilisée ne les supporte pas (interfaces tactiles, [dashboard Mac OS](#) →).

Solution

Afficher les paramètres du logiciel dans un écran "du contenu" plutôt que dans une fenêtre à part. Donner à cet écran les dimensions de la fenêtre du logiciel.

Utiliser une transition, des visuels ou un code couleur pour créer le sentiment que le Backstage soit :

- Un mode éphémère et différent du fonctionnement normal du logiciel,
- Lié exclusivement et directement au logiciel dont il est l'envers,
- Dans lequel on va effectuer des actions potentiellement destructrices qui affecteront l'ensemble du logiciel.

Se référer aux [Indices ambiants](#) pour plus de détails.

Se référer aux [Indices ambiants](#) pour plus de détails.

Comme on dispose de tout l'écran pour afficher de l'information et pas juste d'un panneau restreint il n'est plus nécessaire de masquer puis dévoiler progressivement l'information. On peut utiliser l'espace disponible pour aplatir l'architecture de l'information en échangeant de la profondeur contre de la largeur.

Comme le **Backstage** va remplacer l'objet manipulé, l'utilisateur risque de perdre le contexte de son action. Il pourrait notamment ne plus savoir ce sur quoi il travaillait, et devoir interrompre sa tâche pour le vérifier. Pire encore, il pourrait en toute bonne foi associer ses actions au mauvais objet et potentiellement perdre des données. Il convient de mettre sur le Backstage un aperçu du contenu manipulé, par exemple avec un [Objet riche](#) ou un [Aperçu](#).

Que mettre dans un Backstage ?

- Droits d'accès (qui a accès en lecture, en écriture, modifier ces droits...),
- Impression (format de papier, nombre de copies, quelles pages...),
- Meta-données (date de création, date de modification, auteur, poids...)
- Partage & Export (différents formats, différentes qualités, inclusion des contacts du carnet d'adresse...),
- Réglages (activer la correction orthographique, changer la vue par défaut, ajuster des comportements...)
- Statistiques (nombre de page, nombre de mots...),
- Versionnage (voir et récupérer des versions antérieures, comparer des versions...).

Backstage partiel (local)

Si les réglages concernent seulement une partie du logiciel, comme une entité sur une carte ou un document parmi plusieurs, alors appliquer le **Backstage** localement, c'est à dire à l'emplacement et aux dimensions de ces objets, plutôt qu'à toute la fenêtre du logiciel.

Contre-emploi

L'efficacité du **Backstage** repose sur la force de la métaphore de « l'envers du décor » : l'opérateur s'attend à y trouver des commandes secondaires ou utilisées peu fréquemment, des réglages ou autres actions "dans les coulisses" qui n'ont pas d'effet direct sur le contenu mais qui servent à faire quelque chose avec.

L'opérateur ne va pas penser naturellement à chercher des commandes de création dans le **Backstage** ; ce n'est donc pas l'endroit où ranger les commandes qu'on a pas eu la place de mettre ailleurs. Dans un tel cas essayez de réorganiser l'architecture de l'information, d'utiliser un [Mega menus déroulant](#) ou peut-être un [ruban](#).

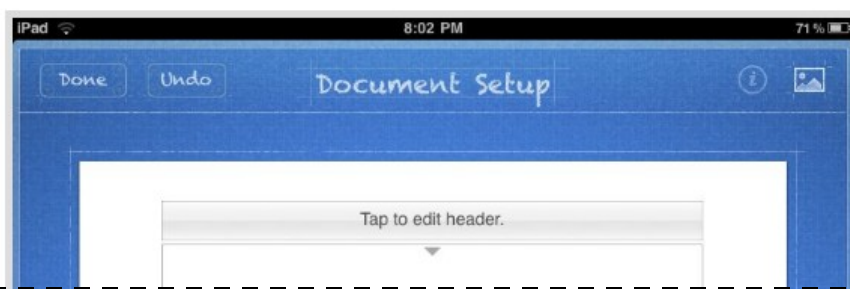
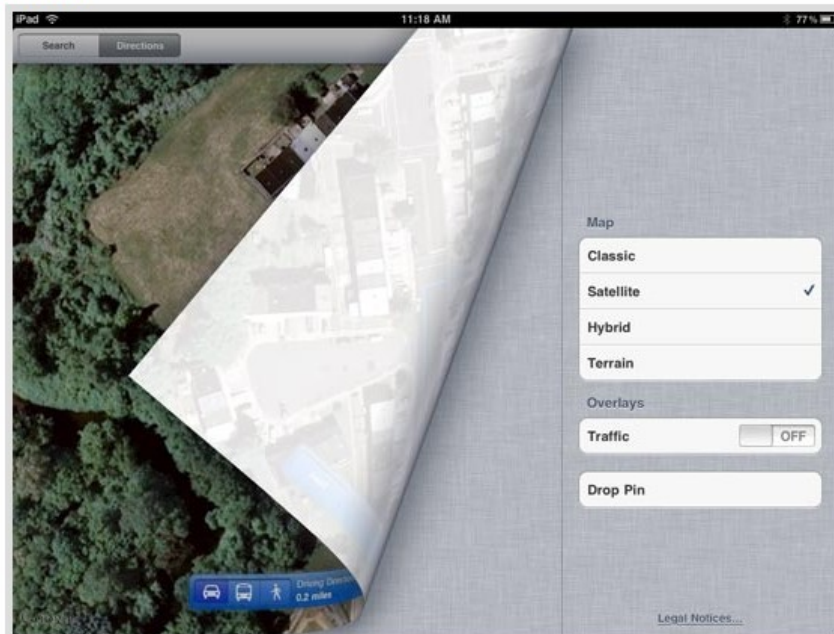
Exemples

Mac OS & iOS

Les widgets Mac OS effectuent une petite animation pour montrer leur *Backstage*.



Les applications iPad ont des *Backstages* magnifiques : soulever un coin de la carte révèle les options possibles de l'application Plans (observez la texture de fond qui à la fois rappelle un matériau de bonne qualité et indique que nous ne sommes pas dans l'environnement "régulier" de l'application). Le mode "Configuration de Document" de Pages ressemble quant à lui à un "blueprint" avec imagerie, icônes, typographie adéquates et spécifiques à cet écran.



C.5 Visualisation des modes de présentation CAE

Bibliothèque de patrons CAE

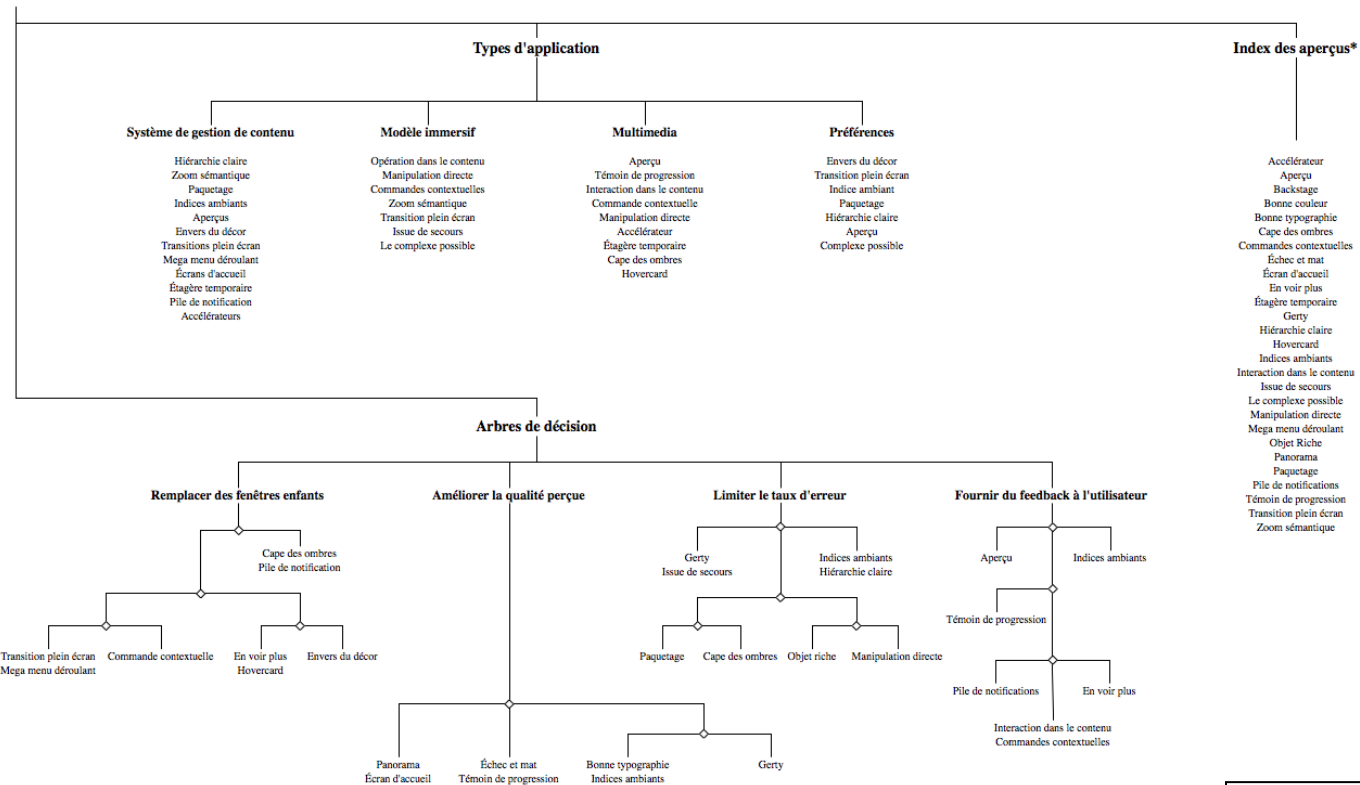


Schéma de patron :

- Aperçu (titre + visuel + description)
- Problème
- Contexte
- Solution
- Exemples
- Pour aller plus loin
- Commentaires (ouverts au public)

ANNEXE D – Évaluation expérimentale

D.1 Certificat d'acceptation d'un projet de recherche par le Comité d'éthique



CERTIFICAT D'ACCEPTATION D'UN PROJET DE RECHERCHE PAR LE COMITÉ D'ÉTHIQUE DE LA RECHERCHE AVEC DES SUJETS HUMAINS DE L'ÉCOLE POLYTECHNIQUE

Comité d'éthique de la recherche avec des sujets humains

Adresse civique :
Campus de l'Université de Montréal
2900, boul. Édouard-Montpetit
École Polytechnique
2500, chemin de Polytechnique
H3T 1J4

Adresse postale :
C.P. 6079, succursale Centre-ville
Montréal (Québec) Canada
H3C 3A7

Téléphone : (514) 340-4990
Télécopieur : (514) 340-4992

École affiliée à
l'Université de Montréal

Membres réguliers du comité :

Marie-Josée Bernardi, juriste et éthicienne
Ginette Denicourt, IRSST
Daniel Imbeau, génie industriel
Bernard Lapierre, éthicien*
Delphine Périé-Cumier, génie mécanique
Élodie Petit, juriste et éthicienne
André Phaneuf, UdeMontréal
Farida Cheriet, génie informatique et génie
logiciel

Céline Roehrig, secrétaire

* président du Comité

Montréal, le 3 mars 2011

M. Jocelyn Richard
M. Jean-Marc Robert
Département de mathématiques et génie industriel
École Polytechnique de Montréal

N/Réf : Dossier CÉR-10/11-09

Messieurs,

J'ai le plaisir de vous informer que les membres du Comité d'éthique de la recherche ont procédé à l'évaluation en comité restreint de votre projet de recherche intitulé « *Création d'une banque de patrons de conception d'interfaces pour logiciels de contrôle de simulateurs de vol et évaluation d'un prototype d'outil de gestion de ces patrons* ».

Les membres du Comité ont recommandé l'approbation de ce projet suite aux précisions que vous avez envoyées par courriel à Mme Roehrig le 25 février, en réponse aux questions soulevées par notre CÉR.

Veuillez noter que le présent certificat est valable pour le projet tel que soumis au Comité d'éthique de la recherche avec des sujets humains. La secrétaire du Comité d'éthique de la recherche avec des sujets humains devra immédiatement être informée de toute modification qui pourrait être apportée ultérieurement au protocole expérimental, de même que de tout problème imprévu pouvant avoir une incidence sur la santé et la sécurité des personnes impliquées dans le projet de recherche (sujets, professionnels de recherche ou chercheurs).

Nous vous prions également de nous faire parvenir un bref **rapport annuel** ainsi qu'un avis à la fin de vos travaux.

Je vous souhaite bonne chance dans vos travaux de recherche,

Bernard Lapierre, président
Comité d'éthique de la recherche avec des sujets humains

c.c.: Céline Roehrig, DRI

D.2 Questionnaire biographique

1. Quel est votre poste chez CAE ?
2. Quelle est votre formation académique ?
3. Avez vous déjà entendu parler de patrons de conception de Gamma (Design Patterns: Elements of Reusable Object-Oriented Software) ? Si oui, où : à l'école, chez CAE, par goût personnel, autre ?
4. Avez vous déjà entendu parler de patrons de conception d'interfaces utilisateur (Design Patterns: Elements of Reusable Object-Oriented Software) ? Si oui, où : à l'école, chez CAE, par goût personnel, autre ?
5. Pourriez-vous évaluer sur une échelle de 1 à 5 votre intérêt à concevoir des interfaces utilisateur ? 1 étant « Je le fais parce que j'y suis parfois obligé », 5 étant « Je pratique sur mon temps libre ».
6. Avez-vous des questions avant que l'on commence ?

D.3 Questionnaire post-expérimentation

1. Pourriez-vous évaluer sur une échelle de 1 à 5 votre satisfaction à utiliser chacun des modes de présentation ? 1 étant « Je ne m'en servais plus », 5 étant « Je m'en servais tous les jours ».
2. Quelle est votre mode de présentation préféré parmi les quatre proposés ?
3. En naviguant avec les modes de présentation de patrons, avez-vous vu des relations entre patrons ? Est-ce que cela vous intéresse ?
4. Si vous ne deviez changer qu'une chose dans la bibliothèque, qu'est ce que ce serait ? Et si vous deviez n'en garder qu'une ?
5. Avez-vous d'autres commentaires ?

D.4 Participants

Patrons de Gamma, *Patrons de UI* et *Habilité au UI* correspondent respectivement aux questions 3, 4 et 5 du questionnaire biographique de la page précédente.

Participant	Poste	Âge	Patrons de Gamma	Patrons de UI	Habilité au UI
Numéro 1	Chef architecte logiciel	38	Oui	Non	3
Numéro 2	Scrum master / Coach en méthodologies Agiles	28	Oui	Non	4
Numéro 3	Chef de développement produit	33	Oui	Non	5
Numéro 4	Développeur logiciel	30	Oui	Non	3
Numéro 5	Designer graphique	37	Oui	Non	4
Numéro 6	Chef architecte logiciel	36	Oui	Non	3,5
Numéro 7	Spécialiste des systèmes logiciels	33	Oui	Oui	5
Numéro 8	Développeur logiciel	33	Oui	Non	3
Numéro 9	Chef de développement produit	35	Non	Non	3
Numéro 10	Responsable de l'assurance qualité logicielle	40	Oui	Non	3

D.5 Patrons trouvés

Participant	Écran choisi	Avancement du projet	Index alphabétique	Types d'applications	Arbres de décision	Index des aperçus
Numéro 1	One UI (sélection de projet) <i>Système de gestion de contenu</i>	Prototype	Bonne couleur	Backstage	Indices Ambiants	Bonne typographie
Numéro 2	Owl (sessions enregistrées) <i>Système de gestion de contenu</i>	Lancement	Paquetage	Mega menu déroulant	Hovercard	Témoin de progression
Numéro 3	AS332 Ownship Status <i>Tableau de bord</i>	Mature	Commandes contextuelles	Zoom sémantique	-	En voir plus
Numéro 4	Build Farm (page produit) <i>Application Web</i>	Prototype	Témoin de progression	Transition plein écran	Pile de notifications	Gerty
Numéro 5	Owl (écran d'accueil) <i>Écran d'accueil</i>	Lancement	Objet riche	Bonne typographie	-	Hovercard
Numéro 6	Build Farm (page produit) <i>Application Web</i>	Prototype	Bonne typographie	Paquetage	Panorama	Le complexe possible
Numéro 7	Owl (scheduler, vue par mois) <i>Calendrier</i>	Lancement	Aperçu	Accélérateurs	Pile de notifications	Hovercard
Numéro 8	Build Farm (page produit) <i>Application Web</i>	Prototype	Bonne couleur	-	Témoin de progression	En voir plus
Numéro 9	Simfinity B737 virtual cockpit <i>Modèle tridimensionnel interactif</i>	Mature	Écran d'accueil	Témoin de progression	Cape des ombre	Panorama
Numéro 10	QA Model Tool (vue audit) <i>Succession de formulaires</i>	Prototype	Bonne couleur	Indices ambients	Cape des ombre	Accélérateurs

D.6 Ordre du choix des modes de présentation par les participants

Participant	1 ^{re} tentative	2 ^{me} tentative	3 ^{me} tentative	4 ^{me} tentative
Numéro 1	Arbres de décision	Index des aperçus	Types d'applications	Index de référence
Numéro 2	Types d'applications	Arbres de décision	Index des aperçus	Index de référence
Numéro 3	Arbres de décision	Index des aperçus	Index de référence	Types d'applications
Numéro 4	Arbres de décision	Types d'applications	Index des aperçus	Index de référence
Numéro 5	Arbres de décision	Index des aperçus	Types d'applications	Index de référence
Numéro 6	Index des aperçus	Arbres de décision	Types d'applications	Index de référence
Numéro 7	Arbres de décision	Types d'applications	Index des aperçus	Index de référence
Numéro 8	Arbres de décision	Types d'applications	Index de référence	Index des aperçus
Numéro 9	Arbres de décision	Types d'applications	Index des aperçus	Index de référence
Numéro 10	Types d'applications	Arbres de décision	Index des aperçus	Index de référence

D.7 Pertinence des patrons trouvés

Calculée en multipliant la spécificité des patrons telle qu'indiquée dans le tableau *Liste des patrons CAE* (annexe D) par 1 si le patron est applicable à l'écran choisi ou par 0 s'il ne l'est pas.

Participant	Index alphabétique	Types d'applications	Arbres de décision	Index des aperçus
Numéro 1	1	3	1	1
Numéro 2	1	2	2	1
Numéro 3	0	3	-	1
Numéro 4	1	3	2	2
Numéro 5	0	1	-	2
Numéro 6	1	1	2	1
Numéro 7	1	2	2	2
Numéro 8	1	-	1	1
Numéro 9	3	1	3	2
Numéro 10	1	1	3	2
Moyenne μ	1.0	1.9	2.0	1.5
Écart-type σ	0.8	0.9	0.8	0.5

D.8 Taux de satisfaction & mode de présentation favori des participants

Participant	Index alphabétique	Types d'applications	Arbres de décision	Index des aperçus	Méthode favorite
Numéro 1	1	4	3	5	Index des aperçus
Numéro 2	1	3	4	5	Index des aperçus
Numéro 3	2	4	3	4	Types d'applications
Numéro 4	4	2	3	5	Index des aperçus
Numéro 5	4	4	4	5	Index des aperçus
Numéro 6	2	3	4	5	Index des aperçus
Numéro 7	4	4	4	4	Index des aperçus
Numéro 8	3	3	4	5	Index des aperçus
Numéro 9	4	5	5	4	Types d'applications
Numéro 10	1	5	3	3	Types d'applications
Moyenne μ	2,6	3,7	3,7	4,5	
Écart-type σ	1,35	0,95	0,67	0,71	

D.9 Temps de recherche pour trouver un patron approprié

Participant	Index alphabétique	Types d'applications	Arbres de décision	Index des aperçus
Numéro 1	02:17	02:12	01:17	01:54
Numéro 2	01:24	04:26	00:47	05:00
Numéro 3	03:06	05:08	03:57	00:50
Numéro 4	07:37	05:47	08:33	03:46
Numéro 5	05:55	07:48	02:29	00:47
Numéro 6	02:45	02:25	02:45	03:17
Numéro 7	01:43	03:00	06:07	04:31
Numéro 8	03:12	03:47	03:59	02:17
Numéro 9	01:53	08:56	03:53	05:36
Numéro 10	01:03	03:21	04:13	06:17
Moyenne μ	03:06	04:41	03:48	03:26
Écart-type σ	02:06	02:16	02:16	01:57

D.10 variation de la satisfaction des utilisateurs relativement à la variation du temps de recherche d'un patron approprié

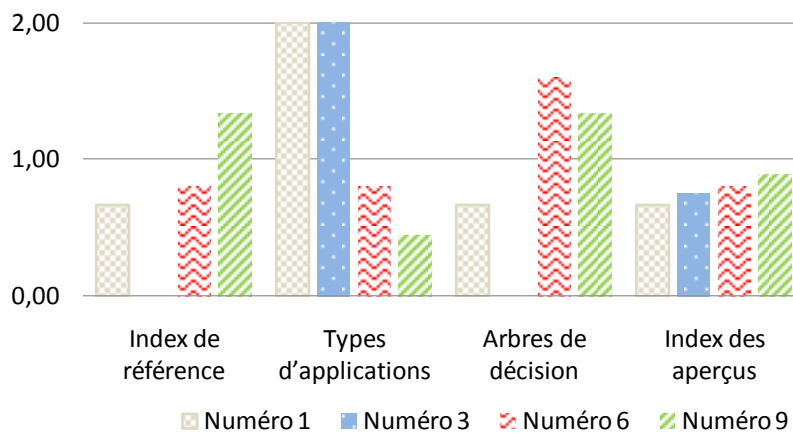
Index de référence	Types d'applications	Arbres de décision	Index des aperçus
1	0,94	1,16	1,65

D.11 Normalisation de la pertinence des patrons trouvés et corrélation avec les catégories de participants

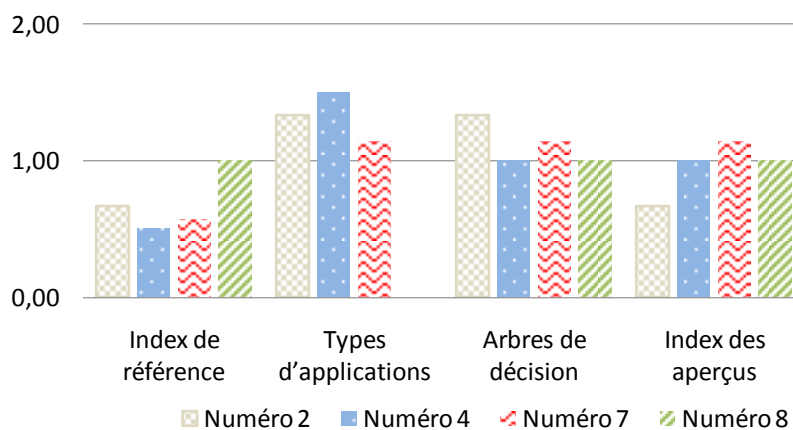
Les résultats précédents ont été divisés par la moyenne pour ramener la dispersion au même ordre de grandeur et la rendre indépendante des unités de mesure. Cette normalisation vise à révéler les éventuelles tendances dans la série de données.

Pertinence	Moyenne μ	Index de référence	Types d'applications	Arbres de décision	Index des aperçus
Numéro 1	1.50	0.67	2.00	0.67	0.67
Numéro 2	1.50	0.67	1.33	1.33	0.67
Numéro 3	1.33	0.00	2.25	0.00	0.75
Numéro 4	2.00	0.50	1.50	1.00	1.00
Numéro 5	1.00	0.00	1.00	0.00	2.00
Numéro 6	1.25	0.80	0.80	1.60	0.80
Numéro 7	1.75	0.57	1.14	1.14	1.14
Numéro 8	1.00	1.00	0.00	1.00	1.00
Numéro 9	2.25	1.33	0.44	1.33	0.89
Numéro 10	1.75	0.57	0.57	1.71	1.14

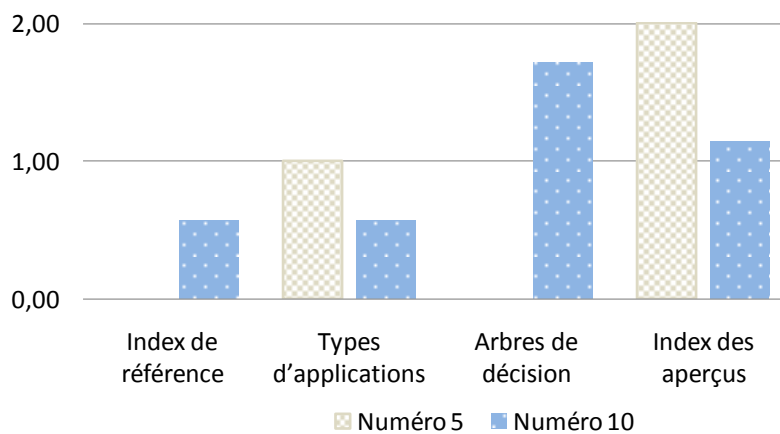
Comparaison de la pertinence des patrons normalisée des développeurs logiciels



Comparaison de la pertinence des patrons normalisée des gestionnaires de produits



Comparaison de la pertinence des patrons normalisée des autres participants

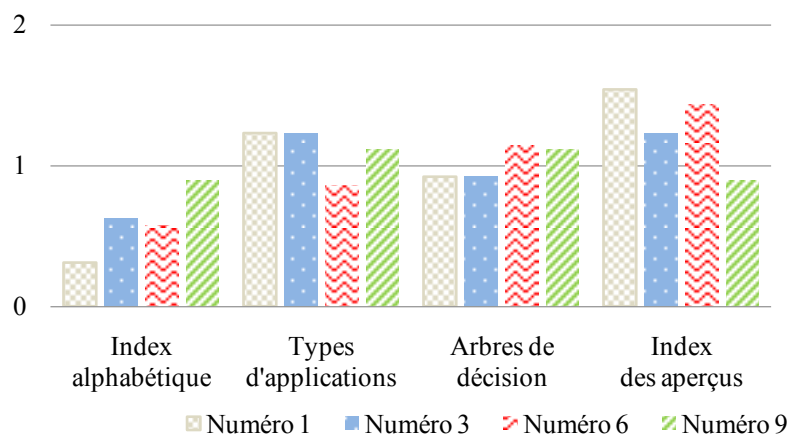


D.12 Normalisation du taux de satisfaction des participants et corrélation avec les catégories de participants

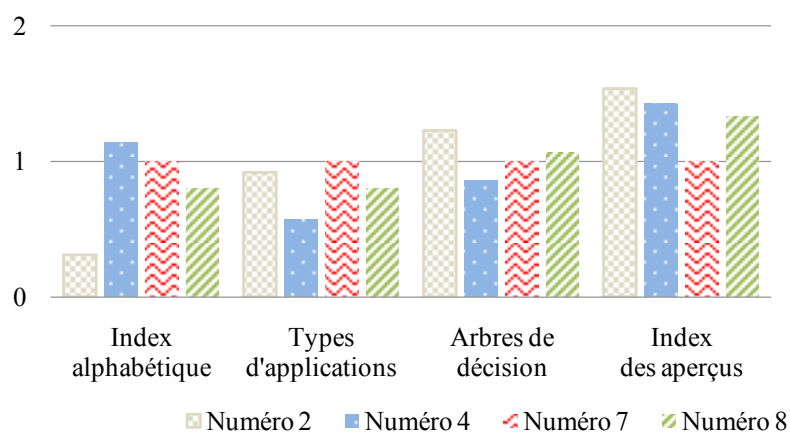
Les résultats précédents ont été divisés par la moyenne pour ramener la dispersion au même ordre de grandeur et la rendre indépendante des unités de mesure. Cette normalisation vise à révéler les éventuelles tendances dans la série de données.

Participant	Moyenne μ	Index alphabétique	Types d'applications	Arbres de décision	Index des aperçus
Numéro 1	3,25	0,31	1,23	0,92	1,54
Numéro 2	3,25	0,31	0,92	1,23	1,54
Numéro 3	3,25	0,62	1,23	0,92	1,23
Numéro 4	3,5	1,14	0,57	0,86	1,43
Numéro 5	4,25	0,94	0,94	0,94	1,18
Numéro 6	3,5	0,57	0,86	1,14	1,43
Numéro 7	4	1	1	1	1
Numéro 8	3,75	0,8	0,8	1,07	1,33
Numéro 9	4,5	0,89	1,11	1,11	0,89
Numéro 10	3	0,33	1,67	1	1

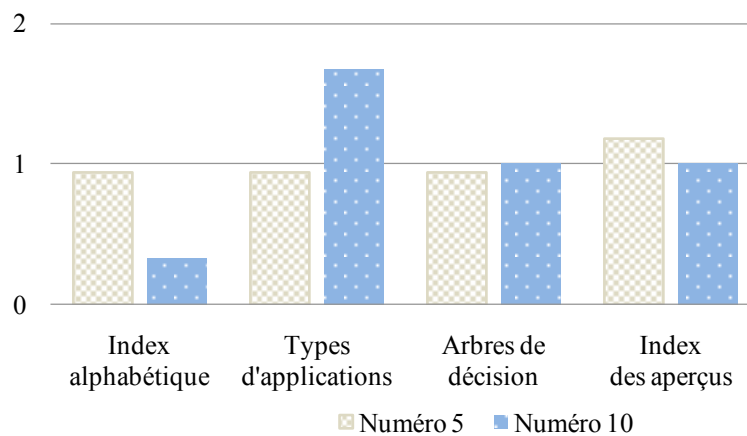
Comparaison des taux de satisfaction normalisés des développeurs logiciels



Comparaison des taux de satisfaction normalisés des gestionnaires de produits



Comparaison des taux de satisfaction normalisés des autres participants

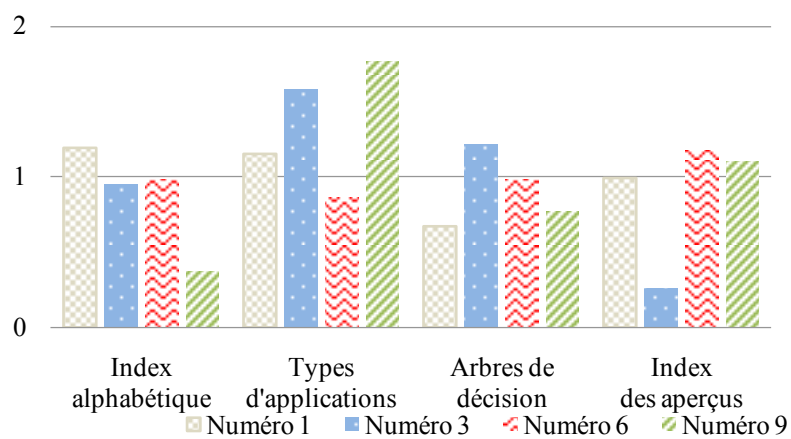


D.13 Normalisation du temps de recherche des patrons et corrélation avec les catégories de participants

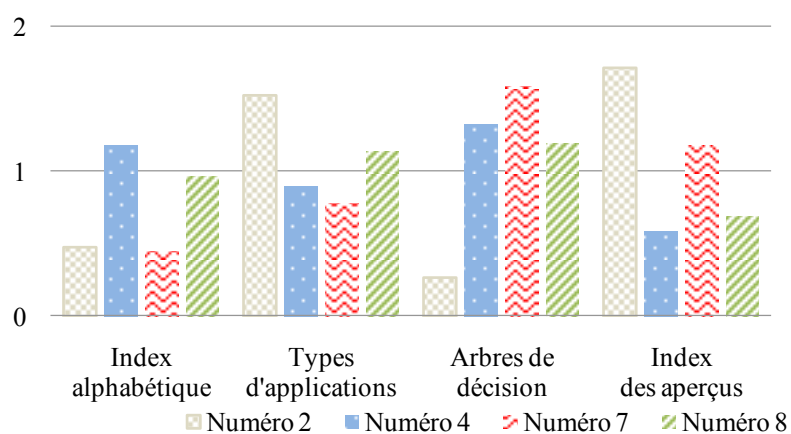
Les résultats précédents ont été divisés par la moyenne pour ramener la dispersion au même ordre de grandeur et la rendre indépendante des unités de mesure. Cette normalisation vise à révéler les éventuelles tendances dans la série de données.

Participant	Moyenne μ	Index alphabétique	Types d'applications	Arbres de décision	Index des aperçus
Numéro 1	01:55	1,19	1,15	0,67	0,99
Numéro 2	02:54	0,48	1,53	0,27	1,72
Numéro 3	03:15	0,95	1,58	1,21	0,26
Numéro 4	06:26	1,18	0,9	1,33	0,59
Numéro 5	04:15	1,39	1,84	0,58	0,18
Numéro 6	02:48	0,98	0,86	0,98	1,17
Numéro 7	03:50	0,45	0,78	1,59	1,18
Numéro 8	03:19	0,97	1,14	1,2	0,69
Numéro 9	05:05	0,37	1,76	0,77	1,1
Numéro 10	03:44	0,28	0,9	1,13	1,69

Comparaison des temps de recherche normalisés des développeurs logiciels



Comparaison des temps de recherche normalisés des gestionnaires de produits



Comparaison des temps de recherche normalisés des autres participants

