



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

L'évaluation ergonomique des didacticiels

Detry, Caroline

Award date:
1996

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

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

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

Take down policy

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

Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique,
Rue Grandgagnage,21
B 5000 - NAMUR (Belgium)

L'évaluation ergonomique des didacticiels

Mémoire présenté en vue de l'obtention
du grade de Licencié et Maître en Informatique

Caroline Detry

Promoteur: Monique Noirhomme-Fraiture

Année académique 1995-1996

lbs 6843725

307327

Je remercie vivement toutes les personnes qui ont de près ou de loin contribué à la réalisation de ce travail.

En particulier :

Madame Monique Noirhomme, Professeur à l'Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix, qui a organisé le stage de fin d'étude et qui a assuré la direction de ce mémoire;

Messieurs Laurent Toutain et Pierre Rolin, qui ont supervisé mon travail lors du stage, pour les conseils et le temps qu'ils ont eu la gentillesse de me consacrer. Je tiens également à remercier les membres de l'Ecole Nationale Supérieure de Télécommunications de Rennes, pour leur soutien et leur disponibilité tout au long de mon stage;

Monsieur Jean Donnay, du département Education et Technologie des Facultés Universitaires Notre-Dame de la Paix, pour sa disponibilité et la documentation fournie;

Mes derniers remerciements s'adressent à mes amis, ma famille et mes proches, sans lesquels tout ce travail n'aurait pas été possible.

Résumé

Dans ce mémoire, nous abordons l'ergonomie des interfaces homme-machine appliquée à l'Enseignement Assisté par ordinateur (E.A.O.). Pour ce faire, nous présentons quelques concepts théoriques relatifs à l'E.A.O. Nous décrivons ensuite les différentes techniques d'évaluation et les intégrons dans une méthodologie de développement de didacticiels. Nous terminons par la critique de la méthodologie employée lors de l'évaluation du «Cours Hypermédia de Langage C». Nous espérons que les résultats obtenus pourront être utilisés à l'avenir dans le développement de nouvelles applications didactiques.

Abstract

In this master thesis, we approach usability of man machine interface applied to Computer Aided Teaching (C.A.T.). We introduce some C.A.T. theoretical notions. We then describe the different ways of evaluating and we integrate them in a methodology for the development of a C.A.T. application. Finally, we criticize a way of evaluating used for the «Cours Hypermedia de Langage C». We hope that the results of our master thesis will be used for the development of new C.A.T. applications.

Table des matières

<i>Introduction</i>	1
<i>Chapitre 1 : L'enseignement assisté par ordinateur</i>	3
1. QU'EST-CE QUE L'E.A.O.?	3
1.1 INDIVIDUALISATION ET INTERACTIVITE	3
1.1.1. <i>L'individualisation</i>	3
1.1.2. <i>L'interactivité</i>	3
2. LA PHILOSOPHIE DOMINANTE DANS LE DEVELOPPEMENT DE DIDACTIQUES	9
2.1. L'APPROCHE I.S.D.	9
2.2. EVOLUTION DE L'I.S.D.	10
2.3. APPORTS ET LIMITES DES METHODOLOGIES I.S.D.	13
 <i>Chapitre 2 : Un nouveau cas : les didactiques hypermédias</i>	 15
1. HYPERTEXTE ET HYPERMEDIA : QU'EST-CE-QUE C'EST?	15
1.1. DEFINITION DE L'HYPERTEXTE	15
1.2. DEFINITION DE L'HYPERMEDIA	17
2. LES PRINCIPALES CARACTERISTIQUES DE L'HYPERMEDIA	17
2.1. LES MODULES	18
2.2. LES LIENS	18
2.3. LA STRUCTURE D'ORGANISATION	19
2.4. LA BASE DE CONNAISSANCES	19
2.5. L'INTERACTIVITE	20
2.6. LES CHEMINS	21
2.7. ANNOTATION ET COLLABORATION	21
3. HYPERMEDIA ET APPRENTISSAGE	22
3.1. CHAMPS D'APPLICATION DE L'ENSEIGNEMENT HYPERMEDIA	22
3.1.1. <i>La recherche d'informations</i>	23
3.1.2. <i>L'acquisition de connaissances du type expertise</i>	23
3.1.3. <i>La résolution de problèmes</i>	24
3.2. INFLUENCES SUR LES ELEVES	24
3.2.1. <i>L'accès</i>	25
3.2.2. <i>Le contrôle</i>	25
3.2.3. <i>La collaboration</i>	26
3.3. INFLUENCE SUR LES CONCEPTEURS	26
3.3.1. <i>Les changements organisationnels</i>	27
3.3.2. <i>La nouveauté du support</i>	27
3.3.3. <i>La tension philosophique</i>	28
3.4. IMPACT SUR LES METHODES D'EVALUATION	28
3.4.1. <i>La richesse de l'environnement</i>	29
3.4.2. <i>Le manque d'expérience des concepteurs</i>	30
3.4.3. <i>Le peu de littérature de référence</i>	30

Chapitre 3 : Apports à l'E.A.O. des philosophies de développement

<i>de logiciels</i>	33
1. LE CENTRAGE SUR LES UTILISATEURS ET LEURS TACHES	33
1.1. CONNAITRE L'UTILISATEUR	33
1.1.1. <i>Définition</i>	34
1.1.2. <i>Impacts possibles sur l'E.A.O.</i>	35
1.2. LA CONCEPTION PARTICIPATIVE.....	35
1.2.1. <i>Définition</i>	35
1.2.2. <i>Impacts possibles sur l'E.A.O.</i>	36
2. LA RECONNAISSANCE DES MERITES DE L'EVALUATION CONTINUE	36
2.1. AVENIR DU CONCEPT D'EVALUATION CONTINUE	36
2.1.1. <i>Les économies possibles</i>	37
2.1.2. <i>Le souci de l'image de marque</i>	37
2.1.3. <i>Le soutien de la littérature et des clients</i>	38
2.2. LES MOYENS DE LA MISE EN OEUVRE DE CETTE VISION	38
2.2.1. <i>Intégration dans le cycle de développement du produit</i>	39
2.2.2. <i>Classification des méthodes d'évaluation</i>	39
2.3. IMPACTS SUR L'E.A.O.....	40

Chapitre 4 : Les évaluations empiriques

1. LES DIFFERENTS TYPES DE TESTS	42
1.1. LES TESTS EXPLORATOIRES	43
1.2. LES TESTS D'EVALUATION	44
1.3. LES TESTS DE VALIDATION.....	44
1.4. LES TESTS DE COMPARAISON	45
2. CAPTURE DES DONNEES DE L'EVALUATION	45
2.1. LES METHODES PARTICIPATIVES.....	46
2.1.1. <i>Le recueil de l'expertise</i>	46
2.1.2. <i>L'analyse des protocoles</i>	46
2.1.3. <i>L'analyse des incidents critiques</i>	47
2.1.4. <i>La construction d'arbre des causes</i>	47
2.2. LES METHODES NON-INTRUSIVES	47
2.2.1. <i>L'enregistrement vidéo</i>	48
2.2.2. <i>Les traces écrites du travail</i>	48
2.2.3. <i>Les questionnaires</i>	48
2.2.4. <i>Les mouchards électroniques</i>	49
2.2.5. <i>Les tactiques visuelles</i>	49
3. DESCRIPTION DES DIFFERENTES ETAPES DE REALISATION D'UN TEST ..	49
3.1. LE DEVELOPPEMENT DU PLAN DE TEST	49
3.2. LA SELECTION DES PARTICIPANTS.....	50
3.2.1. <i>Classement des utilisateurs par catégories</i>	51

3.2.2. Nombre de participants	51
3.3. LA PREPARATION DES MATERIAUX DE TEST	52
3.3.1. Le script d'orientation	52
3.3.2. Le questionnaire de background.....	52
3.3.3. Les instrument de récolte de données.....	53
3.3.4. Le questionnaire de pré-test	53
3.3.5. Les scenarii des tâches	54
3.3.6. Les matériaux nécessaires pour donner l'enseignement pré-requis.....	54
3.3.7. Le questionnaire de post-test	55
3.3.8. Les sujets de débriefing	55
3.4. LA CONDUITE DU TEST	55
3.4.1. La première check-list.....	56
3.4.2. La seconde check-list.....	56
3.4.3. La troisième check-list.....	57
3.5. LE DEBRIEFING DES PARTICIPANTS	57
3.5.1. Le replay test.....	58
3.5.2. La révision des conceptions alternatives	58
3.5.3. La technique du "De quoi vous souvenez-vous?"	58
3.5.4. La technique du "Comment auriez-vous fait?"	59
3.5.5. La technique de l'avocat du diable	59
3.6. LA TRANSFORMATION DES DONNEES RECUEILLIES EN DECOUVERTES ET RECOMMANDATIONS	59
3.6.1. Compilation et résumé des données	60
3.6.2. L'analyse des données	61
3.6.3. Le développement de recommandations.....	62

Chapitre 5 : Les autres méthodes d'évaluation ergonomique 63

1. LES EVALUATIONS INFORMELLES	63
1.1. LES EVALUATIONS HEURISTIQUES	63
1.1.1 La méthodologie de travail.....	64
1.1.2. Nombre et caractéristiques des évaluateurs	64
1.2. LES GUIDELINES	65
1.3. LES PROMENADES PLURIDISCIPLINAIRES	66
1.4. LES INSPECTIONS DE COHERENCE.....	66
1.5. LES INSPECTIONS DE STANDARDS.....	66
1.6. LES PROMENADES COGNITIVES	67
1.6.1. La définition des inputs de la promenade	67
1.6.2. Le rassemblement des analystes.....	68
1.6.3. Déroulement de la promenade.....	68
1.6.4. La capture des informations critiques.....	68
1.6.5. La révision de l'interface	69
1.7. LES INSPECTIONS ERGONOMIQUES FORMELLES	69
1.7.1. L'équipe d'inspection.....	69
1.7.2. Le processus d'inspection	69
1.8. LES INSPECTIONS DE CARACTERISTIQUES	72
2. LES EVALUATIONS FORMELLES	73
3. LES EVALUATIONS AUTOMATIQUES	73

3.4.1. Les systèmes d'évaluation automatique de l'affichage.....	74
3.4.2. Les systèmes de génération automatique de l'affichage.....	75

Chapitre 6 : Comparaison des méthodes d'évaluation de logiciels..... 77

1. COMPARAISON DES METHODES INFORMELLES	78
1.1 LES CRITERES D'EVALUATION DE KARAT	78
1.1.1. La manière de travailler des évaluateurs	79
1.1.2. Les compétences des évaluateurs	79
1.1.3. La façon d'explorer l'interface.....	80
1.2. APPLICATION DES CRITERES AUX PROMENADES PLURIDISCIPLINAIRES, COGNITIVES, AUX EVALUATIONS HEURISTIQUES ET AUX INSPECTIONS ERGONOMIQUES FORMELLES.....	80
1.3. APPLICATION DES CRITERES AUX INSPECTIONS DE STANDARDS	85
ET INSPECTIONS DE COHERENCE.....	85
2. COMPARAISON DES METHODES D'EVALUATION EMPIRIQUES ET DES METHODES D'EVALUATION INFORMELLES	87
2.1. LES LIMITES DE L'EVALUATION EMPIRIQUE	87
2.2. LES LIMITES DE L'EVALUATION INFORMELLE.....	89
2.2.1. Définition des objectifs d'évaluation.....	89
2.2.2. Nombre et type d'erreurs trouvées	89
2.2.3. Génération de recommandations.....	90
2.2.4. Evaluation du produit avant sa diffusion.....	90
3. INTEGRATION DES DIFFERENTES METHODES D'EVALUATION AU COURS DU CYCLE DE DEVELOPPEMENT.....	91

Chapitre 7 : Développement et évaluation de didacticiels..... 97

1. INTEGRATION DES METHODES D'EVALUATION DE LOGICIELS DANS LES ETAPES DE CONCEPTION D'UN DIDACTICIEL CLASSIQUE	97
1.1. LA METHODOLOGIE CLASSIQUE.....	98
1.1.1. Analyse pédagogique.....	99
1.1.2. Validation de l'analyse pédagogique	100
1.1.3. Réalisation de la maquette papier	100
1.1.4. Validation de la maquette papier.....	101
1.1.5. Médiatisation	101
1.1.6. Validation informatique.....	101
1.1.7. Tests.....	101
1.1.8. Corrections	101
1.1.9. Diffusion	102
1.1.10 Evaluation.....	102
1.2. INTEGRATION DE L'EVALUATION ERGONOMIQUE ET DE CONCEPTS ISSUS DU GENIE LOGICIEL	102
1.2.1. Les apports méthodologiques	104
1.2.2. Les apports évaluatifs.....	104
2. LE CAS PARTICULIER DES DIDACTICIELS HYPERMEDIA	107

2.1. AJOUTS LORS DE LA MISE AU POINTS DE DIDACTICIELS.....	107
2.1.1. <i>Le supra système</i>	108
2.1.2. <i>Le système instructionnel</i>	108
2.1.3. <i>Les composants du système instructionnel</i>	111
2.1.4. <i>Les matériaux utilisés</i>	112
2.2. EVALUATION DES DIDACTICIELS HYPERMEDIA	112
2.1.2. <i>Evaluation logicielle</i>	113
2.2.2. <i>Evaluation pédagogique</i>	113

Chapitre 8 : Critique de l'évaluation du didacticiel hypermédia

<i>de langage C</i>	115
1. DESCRIPTION DE LA METHODE UTILISEE.....	115
1.1. ANALYSE DE LA TACHE ET DES UTILISATEURS.....	115
1.1.1. <i>L'analyse de la tâche</i>	116
1.1.2. <i>Identification et définition de la population en différents stéréotypes</i>	116
1.1.3. <i>Description ergonomique de la tâche</i>	117
1.1.4. <i>Description du poste de travail</i>	117
1.2. LE LOGICIEL INCRIMINE	118
1.2.1. <i>Une description succincte du logiciel</i>	118
1.2.2. <i>Une découpe du logiciel</i>	118
1.3. ANALYSE FONCTIONNELLE DE L'INTERFACE.....	119
1.3.1. <i>Les moyens d'interaction utilisés</i>	119
1.3.2. <i>Les objets interactifs utilisés</i>	119
1.3.3. <i>Reconnaissance et listage du ou des styles d'interaction utilisés</i>	120
1.4. L'ANALYSE ERGONOMIQUE DE L'INTERFACE.	120
2. CRITIQUE DE LA METHODE UTILISEE.....	121
2.1. ANALYSE DES UTILISATEURS, DE LA TACHE ET DU POSTE DE TRAVAIL	121
2.2. EVALUATION HEURISTIQUE DE L'ERGONOMIE DE L'INTERFACE.....	122
2.3. ABSENCE D'ANALYSE DE LA DOCUMENTATION ET DU MANUEL D'UTILISATION	122
2.4. PRISE EN COMPTE DES ASPECTS HYPERMEDIA DU PRODUIT.....	123
 <i>Conclusions</i>	 125
 <i>Bibliographie</i>	 127

Introduction

De nombreux auteurs insistent de plus en plus sur l'importance d'intégrer dans la mise au point de logiciels plusieurs étapes d'évaluation de l'interface. La concurrence féroce entre les firmes de développement et l'avantage concurrentiel certain que procure un produit de haute qualité, soucieux de ses utilisateurs, ont exacerbé ce besoin de s'assurer de l'ergonomie des logiciels avant leur diffusion.

D'autre part, il y a des chances que cette tendance observée en génie logiciel trouve écho dans le domaine de l'enseignement assisté par ordinateur (ou E.A.O.). En effet, on note déjà la trace dans les différentes méthodologies de conception de didacticiels d'une évolution vers une analyse pédagogique et ergonomique à différentes étapes du cycle de développement.

La tendance générale, du moins dans la littérature, va donc vers un souci accru du bien-être des opérateurs et de la facilité d'utilisation des produits informatiques, ce qui passe obligatoirement par une évaluation poussée de l'interface. Malheureusement, il existe de nombreuses façons de réaliser cette évaluation et les concepteurs sont parfois perdus devant la multiplicité des méthodes, sans savoir laquelle est la plus à même d'être adaptée au cas particulier de leur propre logiciel. Ce problème est particulièrement aigu en E.A.O., où traditionnellement l'évaluation se centrait moins sur l'interface que sur les performances des étudiants. C'est pourquoi nous nous sommes plus spécifiquement concentrés au cours de ce mémoire sur cette discipline, afin de mieux comprendre quels pourraient être pour elle l'impact de ces changements issus du génie logiciel. Pour ce faire, nous avons structuré ce travail de la manière suivante :

Le premier chapitre rappelle en quoi consiste l'enseignement assisté par ordinateur, et montre l'évolution que cette discipline a subie au cours de ces dernières années. Le second chapitre présente les environnements hypermédias, qui seront probablement appelés à soutenir dans le futur de nombreuses applications didactiques.

Le troisième chapitre explique quant à lui l'influence que les nouveautés en matière de conception de logiciels pourraient amener dans le contexte de l'E.A.O.

Le quatrième et cinquième chapitres expliqueront les différentes méthodes permettant de réaliser l'évaluation ergonomique d'une interface, tandis que dans le sixième, nous nous efforcerons de mettre au point une méthodologie de développement de logiciels intégrant le concept d'évaluation continue.

Lors du septième chapitre, nous verrons comment adapter les méthodologies de développement de didacticiels afin de prendre en compte ces aspects d'inspection de l'interface lors de la conception du produit. Nous terminerons enfin par la critique d'une méthode d'évaluation de logiciels appliquée lors de l'analyse de l'interface d'un didacticiel, le Cours Hypermédia de Langage C, lors du chapitre 8.

Chapitre 1 :

L'enseignement assisté par ordinateur

L'enseignement assisté par ordinateur (ou E.A.O.) a fait son apparition depuis quelques années déjà. De nombreux didacticiels ont vu le jour, dans des domaines aussi variés que l'histoire, les langues étrangères ou le fonctionnement d'Internet.

Il va sans dire que le développement de ces logiciels a nécessité la mise au point de méthodologies propres pour pouvoir prendre en compte à la fois les aspects techniques et les aspects éducationnels de ce type de produit. Mais au-delà de la méthode, il y a toujours une certaine vision, une 'philosophie' de conception sur laquelle se base l'équipe de projet pour aborder la mise au point d'un logiciel.

Avant d'aborder la description de la vision dominante en E.A.O., il me semble tout d'abord important de rappeler en quoi consiste exactement l'enseignement assisté par ordinateur.

1. Qu'est-ce que l'E.A.O.?

Nous nous baserons dans cette section sur l'ouvrage de Besnainou, Muller et Thoiun "Concevoir et utiliser un didacticiel" [BESN88], ainsi que sur le mémoire de X. Gobert "Contribution à l'enseignement assisté par ordinateur dans le domaine des télécommunications : un didacticiel pour Internet" [GOBER95].

On peut définir l'enseignement assisté par ordinateur comme "l'enseignement de matières quelconques utilisant l'ordinateur comme moyen de communication privilégié entre l'élève et l'enseignant" [GOBER95, p.3]

La grosse différence entre un cours donné grâce à l'E.A.O. et un cours classique, outre l'aspect purement technique, réside dans la capacité de l'E.A.O. à "tendre au maximum vers un enseignement individualisé" [BESN88, p.19]. En effet, la 'ressource pédagogue', représentée par le système, perd de sa rareté puisqu'elle devient accessible à tous les étudiants

simultanément. Cette situation amène à un changement de préoccupation dans l'enseignement. En E.A.O., par rapport à la situation des cours classiques, l'intérêt se porte davantage sur l'étudiant que sur la pédagogie elle-même.

"Une pédagogie centrée sur l'élève prend en compte les motivations de celui-ci: ses intérêts, ses goûts, son rythme de travail, son expérience antérieure, etc... C'est donc une pédagogie de l'apprentissage individualisé" [GOBER95, p.4].

"Une pédagogie centrée sur le contenu, en revanche, est fondée sur la manière d'enseigner et non sur l'apprentissage de l'étudiant. Elle est, par définition, centrée sur l'étudiant et la transmission des connaissances" [GOBER95, p.4].

1.1 Individualisation et interactivité

L'E.A.O. est une forme de pédagogie fort tournée vers l'élève. Mais encore faut-il faire participer l'étudiant, lui poser des questions et s'assurer qu'il a compris. Ces deux idées, d'une part un enseignement individualisé et d'autre part un enseignement interactif, sont fondamentales en E.A.O.

1.1.1. L'individualisation

Une individualisation de l'enseignement sous-entend la capacité de prendre en compte les caractéristiques propres à l'étudiant lors de l'apprentissage. Dans un enseignement classique, elle est assez difficile à atteindre. Comment en effet tenir compte des capacités et des goûts des étudiants lorsqu'il faut donner cours à plusieurs centaines de personnes? On doit bien alors se contenter d'un enseignement centré sur l'élève moyen, standard.

La meilleure des solutions pour atteindre au maximum l'objectif de fournir à chaque étudiant un enseignement personnalisé est, sans nul doute, le recours à un précepteur. Cependant, cette solution n'est pas économiquement viable. L'E.A.O., en mettant l'ordinateur à la disposition de tous les élèves, est sans doute un bon moyen d'atteindre à une certaine individualisation dans les cours donnés.

Rappelons tout d'abord que l'individualisation reprend traditionnellement en E.A.O. les processus suivants [BESN88] :

- L'adaptation du contenu du cours en fonction des capacités et compétences de l'élève. Cela implique bien entendu que l'on puisse juger de ces capacités lors du cours. Le didacticiel se doit donc de poser des questions afin de pouvoir disposer des réponses apportées par les différents élèves tout au long de leur parcours d'apprentissage.
- L'adaptation du temps passé avec les différents élèves en fonction de la vitesse d'apprentissage de chacun de ceux-ci.
- La possibilité de sélectionner un sujet dans un contenu, ce qui permet à l'élève de choisir ce qui l'intéresse dans un cours donné.

L'individualisation a pour conséquence "un glissement vers la situation d'auto-formation" [GOBER95, p.4]. Selon Besnainou, les étudiants devraient donc pouvoir progresser à leur rythme, le cours étant donné en fonction de leurs capacités personnelles. De plus ils devraient pouvoir choisir ce qui les intéresse parmi les sujets abordés dans le didacticiel. Cependant, si les deux premières caractéristiques demandées aux didacticiels sont présentes dans la plupart des cours informatisés, il n'en est pas de même pour la dernière. Dans la plupart des cas, le cours est présenté sous forme linéaire, l'étudiant n'ayant pas la possibilité de choisir vers quelle partie du cours se diriger. Les choses sont néanmoins en train de changer à ce niveau, les didacticiels actuels laissant de plus en plus de liberté aux utilisateurs.

1.1.2. L'interactivité

L'interactivité est la seconde caractéristique essentielle en E.A.O. L'étudiant en effet doit au minimum pouvoir entrer ses réponses dans le système et déterminer quand il est prêt à accéder à d'autres informations. L'étudiant agit donc sur le didacticiel et celui-ci lui fournit une réponse en retour, sous forme de corrections d'exercices, de messages d'erreurs ou de nouvelles matières à apprendre. En E.A.O., cependant, l'interactivité est définie de façon un peu particulière, eu égard au fait que les manifestations interactivité doivent être parfaitement définies avant d'être implémentées.

Ces manifestations interactives sont modélisées en E.A.O. par des "unités d'interaction". L'unité d'interaction est définie comme la structure élémentaire d'échange entre le logiciel et l'utilisateur [BESN88].

Ces unités d'interactions sont disposées tout au long du didacticiel, pour permettre aux étudiants de communiquer avec l'ordinateur. Un didacticiel comprend donc un nombre en général important de ce genre d'unités, placées à des endroits stratégiquement choisis. Ainsi que le fait remarquer X. Gobert, "le devoir du concepteur est de maîtriser cette structure et d'assurer le chaînage des unités entre elles" [GOBER95, p. 5].

Une unité d'interaction peut être considérée comme un ensemble composé d'un minimum de 4 éléments. Mais le nombre de ces éléments varie selon le degré d'interactivité de l'unité. Il s'agit de :

- la sollicitation machine,
- l'entrée des données par l'élève en fonction de la sollicitation,
- la réaction de la machine aux données entrées par l'élève,
- la décision de branchement.

a.) La sollicitation machine

Pour que l'étudiant puisse interagir avec l'ordinateur, il faut encore que le didacticiel soit programmé pour fournir à ses utilisateurs un moyen de lui communiquer des informations. Ces moyens de communiquer peuvent être des boutons, des boîtes à cocher,... Ces moyens de communiquer avec l'ordinateur doivent être placés dans le didacticiel de manière à ce que l'utilisateur puisse réagir à une sollicitation de la machine. Par exemple, l'utilisateur doit avoir la possibilité de répondre à une question posée par la machine, ou entrer des paramètres pour l'initiation d'une simulation dans le cas d'un logiciel conçu sous cette forme.

b.) L'entrée des données par l'élève en fonction de la sollicitation

L'étudiant répond effectivement à la sollicitation machine en cochant le bouton approprié, en entrant des données alphanumériques,...

c.) La réaction de la machine aux données entrées par l'élève

La machine reçoit les informations données par l'élève et les analyse. Selon le type d'informations reçues, la machine va réagir. Par exemple, si la machine avait posé une question à l'élève, elle fera un commentaire sur la réponse de ce dernier. Ou alors elle exécutera la simulation selon les ordres donnés par l'étudiant.

d.) La décision de branchement

La machine ayant analysé les informations que lui a transmis l'élève, elle le dirige vers la prochaine unité d'interaction. Il est à noter que cette unité d'interaction peut-être la même unité d'où l'on vient, notamment en cas de réponse incorrecte de l'étudiant.

Ces quatre éléments s'articulent comme suit :

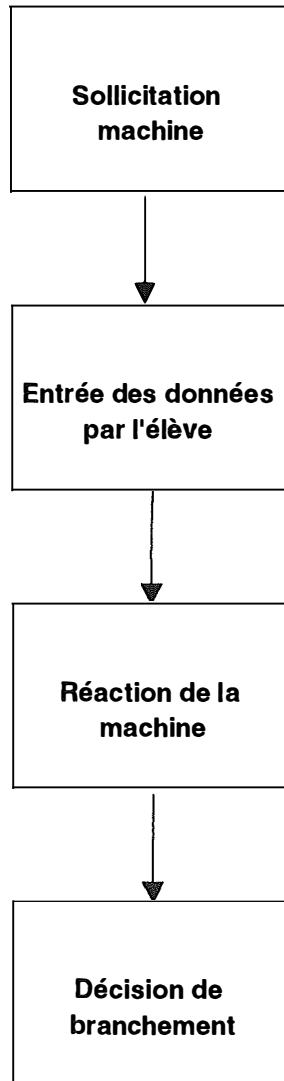


Schéma 1 : Structure de l'unité d'interaction [BESN88, p.22]

La structure décrite n'est pas la seule possible. Au contraire, il est possible pour le concepteur de l'améliorer, de l'enrichir en introduisant entre les deux premières étapes les fonctions suivantes :

- l'appel d'aide contextuelle.
- l'appel d'une fonction de calcul.
- le refus par l'élève de la sollicitation machine.
- le retour en arrière.

Des unités sont dites non-interactives si elles ne sont composées que de la première et de la quatrième étapes. Il s'agit typiquement d'un écran à lire, d'une séquence animée à regarder,...

Le caractère interactif d'un didacticiel se mesure au nombre d'unité d'interaction qu'il contient. Au plus un didacticiel comprend d'unité interactives, au plus il est dit interactif.

2. La philosophie dominante dans le développement de didacticiels

Il n'y a pas eu qu'une évolution pédagogique au cours de ces dernières années. Certes, l'E.A.O. a recentré les préoccupations sur l'étudiant, ce qui en soi constitue déjà un changement important. Cependant, voir l'E.A.O. comme un mode de pensée monolithique consisterait en une grave erreur. En effet, l'E.A.O. est traversé de différents courants, issus du domaine de l'éducation, du domaine de l'intelligence artificielle ou de la recherche.

2.1. L'approche I.S.D.

L'Instructional System Development (ou I.S.D.), décrit par Logan dans son ouvrage "Instructional System Development, An Internal View of Theory and Practice" [LOGAN82] est une approche assez ancienne mais qui est toujours largement utilisée aujourd'hui. Elle nous vient en droite ligne de la cybernétique et est basée sur le modèle mis au point par Banathy, Churchman et von Bertalanffy (cfr. schéma 2).

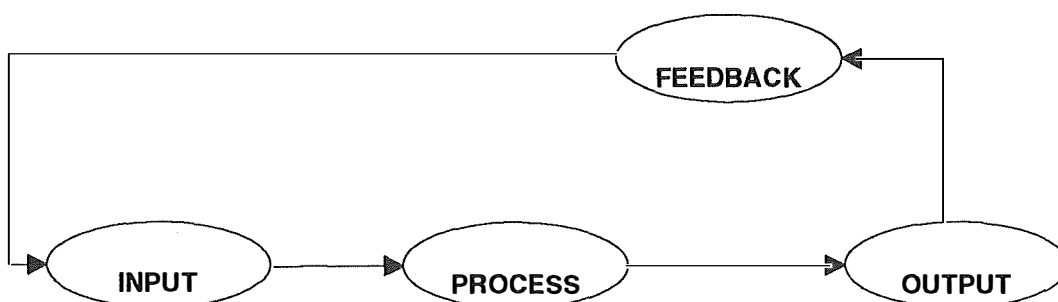


Schéma 2 : Le modèle cybernétique [LOGAN82, p.4]

L'I.S.D. se base sur la définition d'un certain nombre de résultats que l'on désire voir atteindre par l'utilisateur. Ces résultats constituent les objectifs du cours. Cette définition des objectifs doit avoir lieu très tôt dans le développement du didacticiel. Ils sont décrits sous forme de comportements observables et de performances à atteindre. Des tests sont conçus pour mesurer les performances des étudiants suite au cours. Les tests doivent "être construits dès que les objectifs sont définis et que la matière est analysée, donc avant l'élaboration du programme même" [LECLE77, p.116], dans le but de ne pas se laisser influencer par le programme.

Une fois ces objectifs et les tests pour mesurer leur accomplissement définis, il s'agit pour les concepteurs de produire du matériel didactique qui concevra les 'événements' qui accompliront le résultat souhaité.

Ensuite, un cours est donné sur base du matériel didactique ainsi produit. Les performances des étudiants sont chiffrées, rassemblées et analysées, puis comparées aux objectifs pédagogiques définis lors de la première étape. Le système existant sera alors modifié en fonction de ces analyses.

2.2. Evolution de l'I.S.D.

Au cours de ces dernières années, la philosophie même de l'I.S.D. a connu quelques changements. Ceux-ci se sont marqués dans l'évolution des méthodologies basées sur cette vision. Dans les premiers temps, les méthodologies ne se préoccupaient pas d'effectuer des tests avec les utilisateurs avant que le produit ne soit déjà bien développé. Suivant en cela la vision traditionnelle de l'I.S.D., les tests et révisions du système n'avaient lieu que tardivement, pratiquement à la fin du cycle de développement du produit, lors des mesures de performance des étudiants. Cette approche était très lourde, car si dès le départ le produit était mal pensé, on ne s'en apercevait que lorsqu'il était trop tard que pour pouvoir effectuer des modifications en profondeur à des coûts raisonnables.

Une évolution s'est cependant produite dans la façon de concevoir et développer les didacticiels au cours de ces dernières années. On essaie de minimiser les coûts et les efforts de modifications des didacticiels tout en tentant d'améliorer leur ergonomie. Dans cette optique,

les concepteurs intègrent des feed-back des utilisateurs tout au long du développement du produit, permettant de valider le travail déjà effectué. Ces feed-back sont le plus souvent matérialisés sous forme de tests. Comme le didacticiel n'est pas encore disponible aux premières étapes de développement, les tests sont supportés par des dessins d'écrans ou des prototypes. Cette façon de faire est évidemment intéressante d'un point de vue économique puisque le 're-pensage' du produit se fait étape par étape. Les risques de devoir reprendre le didacticiel de A à Z suite à des performances catastrophiques des utilisateurs s'en retrouvent donc fortement réduits.

Un exemple de cette évolution est donné par la comparaison entre la méthodologie proposée par D. Leclercq, J. Donnay et R. De Bal dans l'ouvrage "Construire un cours programmé" [LECLER77] et celle proposée par R. Besnainou, C. Muller et C. Thouin dans l'ouvrage "Concevoir et utiliser un didacticiel" [BESN88].

La méthode de D. Leclercq est assez ancienne (1977). Le cycle de développement d'un didacticiel est divisé en 7 étapes principales, décrites ci-dessous :

- définition des objectifs, où l'on analyse la population cible en terme de connaissances préalables et où l'on fixe les objectifs à atteindre, c'est-à-dire que l'on détermine quelles seront les connaissances que devront maîtriser les étudiants à la suite du cours et comment s'assurer que ceux-ci les ont bien acquises.
- analyse comportementale des objectifs, où l'on divise la matière à enseigner en modules.
- élaboration des séquences d'apprentissage, où l'on organise ces modules entre eux afin de les présenter dans le meilleur ordre pédagogique possible.
- épreuves d'encadrement, où l'on rédige les tests auxquels les étudiants devront répondre afin de vérifier s'ils ont bien rempli l'objectif fixé.
- choix du type d'itinéraire, où l'on choisit un mode de présentation du programme (par exemple, suit-on un déroulement linéaire ou le programme repose-t-il à l'étudiant une question à laquelle celui-ci aurait mal répondu ?).
- rédaction des items, où l'on écrit à proprement parler les unités d'interaction.
- mise au point expérimentale, où l'on procède à l'ajustement du cours suivant les performances des étudiants.

La méthode de Besnainou est, en revanche, beaucoup plus récente (1988). Le cycle de développement d'un didacticiel est divisé en 10 étapes principales, décrites ci-dessous :

- L'analyse pédagogique, où l'on analyse la population cible en terme de connaissances préalables et où l'on fixe les objectifs à atteindre, c'est-à-dire que l'on détermine quelles seront les connaissances que devront maîtriser les étudiants à la suite du cours et comment s'assurer que ceux-ci les ont bien acquis. On précise en outre le contenu du cours, ses objectifs, les tests pour voir si ces derniers sont bien remplis, la découpe en unités d'interactions et la façon dont celles-ci vont s'enchaîner les unes par rapport aux autres,
- la validation de l'analyse pédagogique, où l'on fait examiner l'analyse pédagogique par le commanditaire du didacticiel,
- la réalisation de la maquette papier, où le didacticiel est entièrement décrit et spécifié,
- la validation de la maquette papier, où la maquette est testée sur un échantillon de la population cible,
- la médiatisation, qui recouvre la conception détaillée et l'implémentation du didacticiel,
- la validation informatique, où l'on teste le programme pour vérifier l'absence de bugs,
- les tests, où le didacticiel est testé sur un échantillon d'utilisateurs afin de vérifier son ergonomie,
- les corrections, où l'on modifie le didacticiel suite aux résultats des tests,
- la diffusion du produit,
- l'évaluation, qui constitue en quelque sorte le suivi du didacticiel diffusé.

L'évolution a donc été très nette entre ces deux approches. Tout d'abord, on peut constater que l'approche préconisée par Besnainou, Muller et Thouin semble davantage tournée vers le support. En effet, elle propose des tests informatiques, l'élaboration d'une maquette papier du didacticiel,... Toutes ces étapes sont absentes de la première méthodologie, qui semble, elle, beaucoup plus centrée sur la stratégie pédagogique, puisqu'elle comprend l'analyse comportementale des objectifs, l'élaboration des séquences d'apprentissage, ou encore la rédaction des épreuves d'encadrement. Cependant, les deux méthodologies ne s'opposent qu'en apparence. Les étapes pédagogiques de D. Leclercq sont en effet reprises dans l'analyse pédagogique de Besnainou. La méthode utilisée par ce dernier est simplement plus axée sur l'informatique puisqu'il a ajouté dans le cycle de développement du didacticiel des

étapes visant à assurer la qualité de l'interface et du logiciel en tant que tel. Il y a donc eu une évolution vers le support logiciel, venant sans doute de la prise de conscience de ce que la qualité de ce dernier influence de façon non négligeable la facilité d'apprentissage de l'étudiant [DUFRESN91].

2.3. Apports et limites des méthodologies I.S.D.

L'approche I.S.D., ou par raffinements successifs, est assez ancienne. Elle est à la base de nombreuses méthodologies encore fort répandues à l'heure actuelle. Quant aux avantages de cette approche, ils sont nombreux. Citons, entre autre :

- La reproductibilité d'un bon système, ce qui signifie qu'une fois la bonne recette trouvée, on peut aisément en reproduire les ingrédients pour d'autres systèmes éducationnels.
- La facilité d'adaptation de l'enseignement à grande échelle, c'est-à-dire qu'une fois tous les ingrédients définis, il devient facile de faire porter l'effort instructionnel à grande échelle.
- Le centrage de la pédagogie sur les performances de l'étudiant, c'est-à-dire que non seulement on retrouve l'avantage des système E.A.O. de pédagogie centrée sur l'étudiant (et plus la pédagogie centrée sur le contenu et le professeur [BESN88], comme c'est le cas dans les cours classiques), mais en plus, il devient très facile de voir si un cours a bien atteint son objectif puisque cette évaluation est réalisée via des performances chiffrées.

Malheureusement, cette vision cybernétique a aussi des revers. Elle peut être mal perçue, de par son caractère assez proche des processus industriels puisqu'elle présente les choses en terme d'audit, de contrôle de qualité...[BESN88]. De plus, elle implique une organisation centralisée en terme de temps et de gens impliqués [LOGAN82].

D'autre part, la définition des objectifs, si importante dans la philosophie I.S.D., peut elle-même poser problème. Tout d'abord, il faut remarquer que tout le monde n'est pas d'accord sur cette façon de mesurer l'efficacité d'un cours. En effet, des objectifs en terme de comportements observables, avec une valeur chiffrée de référence, ne font pas l'unanimité. Selon quel étalon en effet mesurer l'esprit critique, la créativité...? [LECLER77]. Ces qualités

sont, par définition, difficilement mesurables. On doit dès lors se contenter des comportements observables pour vérifier l'efficacité d'un cours, et donc pour définir ses objectifs. Par ailleurs, un autre problème est apparu récemment, lié à l'enseignement hypermédia. Celui-ci résiste en effet à la définition des objectifs préconisée en I.S.D. Nous allons aborder ce problème au cours du chapitre suivant.

Chapitre 2 :

Un nouveau cas: les didacticiels hypermédias

Au cours de ces dernières années, de nouveaux supports, l'hypermédia et l'hypertexte, sont apparus comme support à l'enseignement assisté par ordinateur. Les systèmes de ce type sont, de l'avis de nombreux experts, promis à un bel avenir. Il existe des applications hypermédia avec un jargon éducationnel, comme les versions hypertextes des journaux, encyclopédies...[NIELS89], et même certains produits purement éducationnels ont déjà été créés (par exemple "Elements de Biologie Moléculaire" de Ayissi Etemme [AYISS96]). Cependant, le fait que la littérature soit très dispersée à ce sujet montre que ces concepts n'ont été l'objet de la recherche que depuis quelques années.

En général, une idée communément admise est que l'enseignement hypermédia est assez bien adapté pour un enseignement "ouvert", où l'étudiant est libre de ses actions et encouragé à prendre l'initiative. Nous essaierons de faire un peu le point sur cet aspect des choses. Pour ce faire, nous allons commencer par rappeler les définitions d'hypertexte et hypermédia, leurs caractéristiques, pour finalement aborder les aspects de ces techniques qui semblent intéressants dans le domaine de l'éducation.

1. Hypertexte et Hypermédia : Qu'est-ce-que c'est?

Nous nous baserons pour définir ces systèmes sur l'ouvrage "Hypertexte et hypermédia" de Jakob Nielsen [NIELS89], ainsi que sur le "Dictionnaire du multimédia" de Jacques Notaise, Olivier Dusanter et Jean Barda [NOTAIS95].

1.1. Définition de l'hypertexte

D'après Nielsen [NIELS89], pour comprendre ce qu'est l'hypertexte, la façon la plus simple de procéder est de le comparer avec un texte traditionnel. Le texte, dans sa vision traditionnelle, est constitué d'une séquence de mots, constituant des phrases, qui elles-mêmes constituent un texte. Il n'est pas question dans ce cas de lire une phrase ou un mot avant

l'autre. L'organisation des mots et des phrases suit une logique séquentielle, qu'il faut impérativement suivre sous peine de ne rien comprendre.

Dans le cas de l'hypertexte, il n'existe pas d'ordre déterminé disant ce qu'il convient de lire à la suite de tel fragment de texte. On dit donc que l'hypertexte est organisé de façon non-séquentielle. Bien sûr, il n'est pas possible qu'un texte soit entièrement non-séquentiel. Il existe toujours des morceaux de texte à l'intérieur desquels l'ordre de lecture est séquentiel. Mais ces fragments sont organisés de telle façon que l'on puisse les lire indépendamment les uns des autres, sans ordre prédéterminé. Le schéma ci-dessous montre un exemple de cette organisation. Après avoir lu le fragment A, le lecteur a le choix entre continuer sa lecture via le fragment de texte B, E, ou F, et ainsi de suite.

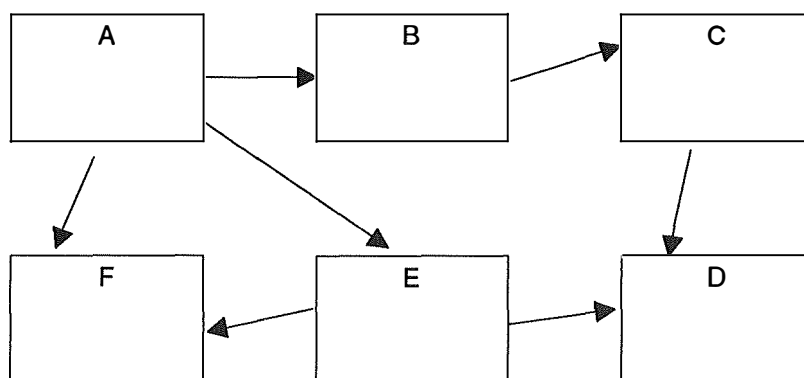


schéma 3 : exemple de structure d'un hypertexte [NIEL89, p.1]

Bien évidemment, cette liberté offerte à l'utilisateur de pouvoir choisir quelle suite donner à sa lecture au moment même où il lit implique pour celui qui a écrit le texte un surcroît de travail. Il ne suffit plus de rédiger un texte, il faut encore l'organiser en fragments indépendants, où le fait de n'avoir pas lu un de ces fragments n'hypothèque pas la compréhension de la suite.

Ce phénomène de choix n'est cependant pas nouveau. Il est en effet déjà apparu dans les livres, où le lecteur est sollicité à choisir entre poursuivre sa lecture ou lire les notes de bas de pages. C'est cependant dans les logiciels hypermédias que l'on retrouve cette liberté de lecture de la façon la plus générale.

1.2. Définition de l'hypermédia

Quand on parle d'hypertexte, on sous-entend un système présentant à ses utilisateurs des informations sous forme textuelle. Les systèmes hypermédias, pour leur part, sont beaucoup plus larges puisqu'ils peuvent contenir non seulement du texte, mais aussi de l'image et du son. Le vocable hypermédia est donc employé pour souligner les aspects multimédias de tel ou tel système (nous vous renvoyons à l'ouvrage de M. Noirhomme et L. Goffinet, "Multimédia, Actes de la journée d'information sur le multimédia", p.21, pour une définition précise du terme multimédia [NOIRH95]). Dans la suite de cet exposé, nous emploierons indifféremment les deux termes puisque tout ce que nous pouvons dire au sujet de l'un est adaptable à l'autre, la seule différence entre les deux techniques résidant, rappelons le, dans la capacité ou non à intégrer de l'image ou du son.

2. Les principales caractéristiques de l'hypermédia

Les caractéristiques les plus importantes de l'hypermédia ne font pas l'unanimité au sein des auteurs. En fait, il semble bien que chacun d'eux ait une vision un peu différente de ce qu'il convient de regrouper sous ce vocable. Il est donc assez difficile d'en donner un aperçu suffisamment large pour y englober les multiples courants de pensées compris dans la littérature, et en même temps suffisamment précis que pour ne pas tomber dans le piège d'une vision trop réductrice. Dans cet exposé, le parti a été pris de présenter les caractéristiques les plus générales, c'est-à-dire celles qui se rattachent à la plupart des systèmes hypermédias. Il convient néanmoins d'insister sur le fait que certains systèmes ne répondent pas à la totalité de ces caractéristiques, sans que leur appartenance à la famille hypermédia puisse être mise en doute pour autant.

Nous nous baserons pour décrire les principales caractéristiques de l'hypermédia sur l'article de David Jonassen et R. Grabinger [JONAS90], ainsi que sur l'ouvrage de Jakob Nielsen "Hypertext and Hypermédia" [NIELS89].

D'après David Jonassen, on distingue 7 caractéristiques principales dans les systèmes hypermédias.

Il s'agit de :

- les modules,
- les liens,
- la structure d'organisation,
- la base de connaissance,
- l'interactivité,
- les chemins,
- les annotations et la collaboration.

2.1. Les modules

Ainsi que nous l'avons expliqué, un système hypertexte est composé d'un ensemble de fragments reliés entre eux. Ces fragments, appelés modules, représentent l'unité de base de stockage d'information. Ces modules varient en taille, et peuvent contenir différents types d'informations (sons, animations,...).

Il est intéressant de constater que le contrôle de l'utilisateur sur ces modules varie fortement selon le type de système hypertexte employé. Certains de ces systèmes en effet permettent à l'utilisateur de modifier les modules qui lui sont présentés ou même de créer ses propres modules. D'autres systèmes n'ont pas intégré cette option, les modules "reflétant les exigences en traitement de l'information d'une tâche définie comme but du système" [JONAS90, p.6].

2.2. Les liens

Les modules sont interconnectés entre eux. Ces interconnexions sont appelées liens. C'est à l'aide de ces liens que l'utilisateur peut se déplacer "à travers l'espace d'informations jusqu'aux modules sélectionnés" [JONAS90, p.6]. L'utilisateur qui se déplace se sert de la souris, du clavier, de son doigt dans le cas d'un écran tactile pour indiquer au système le module qu'il désire atteindre.

Les liens ont cependant une autre fonction que celle de fournir un moyen de transport entre les modules. Ils renseignent aussi l'utilisateur sur l'organisation, la nature de la relation entre les différents modules.

Certains systèmes permettent à l'utilisateur de créer ses propres liens ou de modifier les liens existants.

2.3. La structure d'organisation

La façon dont les modules sont reliés montre une certaine logique. "Les modules peuvent être reliés entre eux sur base d'une relation sémantique du sujet concerné ou sur base du traitement de l'information requis par la tâche" [JONAS90, p.6-7]. Bien sûr, dans les cas où l'utilisateur peut lui-même ajouter ou modifier des modules et des liens à l'espace d'informations, on peut dire qu'il effectue sa propre structuration sémantique de l'espace d'informations.

La manière dont l'espace d'informations est structuré, organisé, va avoir une grande influence sur l'utilisateur. Si la manière dont sont organisées les informations est semblable à la façon dont l'utilisateur conçoit les liens sémantiques entre les différentes parties du document hypertexte, alors il lui sera beaucoup plus facile de rechercher des informations dans le logiciel ou de comprendre la matière du cours dans le cas d'un didacticiel.

2.4. La base de connaissances

La tentation est grande de confondre hypermédia et base de données, avec un accès et une recherche de l'information facilités. Du point de vue de la structure d'organisation, il existe indéniablement des similitudes entre de nombreuses bases de connaissances hypermédiées et les bases de données classiques. Pour ajouter à la confusion, il arrive fréquemment que des systèmes hypermédiés se servent de bases de données comme support pour stocker leurs modules.

Mais les ressemblances s'arrêtent là. L'hypertexte en effet n'a que peu de rapport avec les bases de données au point de vue de la structure des informations stockées. Dans les bases

de données, les informations stockées dans une même table ont toutes la même structure, qui a été décidée au préalable et définie par un langage de haut niveau de définition des données" [NIELS89, p.8]. Ainsi que le souligne Nielsen, peu importe que l'on ait des milliers d'employés dans une société, ils sont tous représentés dans la base de données par des informations structurées de façon unique : nom, adresse, salaire...

Dans les systèmes hypermédias, une telle structuration n'existe pas. Les informations constituant la base de connaissances ne sont pas obligatoirement définies de manière centrale. Il n'y a donc pas de structure à appliquer aux données. Pour reprendre l'exemple des employés d'une société, on peut supposer que ces derniers soient représentés dans la base de connaissances par des modules. Ces modules peuvent être l'oeuvre d'un auteur, mais ils peuvent aussi, dans certains systèmes, être conçus par les employés. Les modules seront donc tous différents, selon les goûts de chacun. La façon même dont les modules se structureront entre eux peut également être laissée à l'appréciation des employés. Par exemple, un employé utilisera des liens pour permettre d'accéder de son module à des modules expliquant son travail, ses projets,...

"En général, la structure d'un réseau hypertexte est définie comme l'union des décisions locales prises en construisant chacun des modules d'information et des liens" [NIELS89, p.8]. Cela montre la flexibilité des systèmes hypermédia, avec tous les avantages et les inconvénients que cela implique.

2.5. L'interactivité

L'interactivité est une notion fondamentale en hypermédia. Comme le souligne D. Jonassen, "l'hypermédia permet à l'utilisateur de déterminer sa séquence d'accès à l'information, de faire des ajouts ou des rectifications sur l'information de manière à la rendre plus personnellement significative (collaboration) ou de construire et structurer sa propre base de connaissances" [JONAS90, p.7]. C'est une grande nouveauté par rapport aux systèmes classiques, et les opportunités au niveau de l'E.A.O., qui vise lui aussi une grande interactivité sont importantes. Nous reviendrons sur les apports de l'hypermédia à l'E.A.O. plus loin dans ce chapitre (paragraphe 3). Tous les systèmes ne fournissent cependant pas à l'utilisateur le même degré de liberté et de contrôle. En général, force est de constater que les utilisateurs qui

interagissent avec un système hypermédia ont une difficulté supplémentaire à laquelle faire face. En effet, le fait qu'ils puissent exercer un contrôle sur l'information implique qu'ils ne subissent plus de manière passive le déroulement des séquences d'informations. C'est à eux de déterminer leur propre séquence de traversée de la base de connaissances.

2.6. Les chemins

Les chemins constituent un peu une parade contre la difficulté à se retrouver dans l'espace d'informations, où des utilisateurs peu avertis risquent de se perdre. Il ne s'agit cependant pas de limiter le grand contrôle laissé aux utilisateurs dans l'exploration de la base de connaissances, mais plutôt de leur permettre de voyager en utilisant des visites guidées. Cette façon de visiter linéairement l'espace d'information est souvent employée par les utilisateurs novices.

Les chemins ne sont cependant pas uniquement l'oeuvre du ou des auteurs de la base de connaissances, désireux que les utilisateurs se retrouvent dans l'espace d'information. Dans certains systèmes, ils peuvent également être l'oeuvre des utilisateurs eux-mêmes. Ceux-ci peuvent en effet créer leur propre séquence d'accès en déterminant seuls leurs chemins. De plus, des systèmes sont très souples et permettent de sauvegarder les chemins créés et de les annoter. Dans des systèmes collaborants, les autres utilisateurs ont accès aux chemins ainsi créés.

Les chemins construits par les utilisateurs sont intéressants, car ils montrent comment les utilisateurs perçoivent l'espace des informations et quels sont les liens sémantiques qu'ils effectuent dans les modules.

2.7. Annotation et collaboration

Comme nous l'avons vu, des systèmes hypermédiés permettent de faire partager à tous les utilisateurs les modifications de modules ou de liens effectués par certains. Dans le cas des réseaux, les utilisateurs peuvent bénéficier "d'accès simultanés à l'information dans la base de

connaissances" [JONAS90, p.8]. On parle dans ce cas d'annotation de la base de connaissances et de collaboration.

3. Hypermédia et apprentissage

L'hypermédia va, dans les années à venir, être appelé à soutenir bon nombre d'applications dans le domaine de l'E.A.O. Il faudrait donc dès à présent se pencher sur les défis et opportunités de l'enseignement hypermédia, afin que ces didacticiels soient conçus en connaissance de cause. En effet, toutes les influences de l'hypermédia au niveau des étudiants et des concepteurs de matériel didactique ne semblent pas encore clairement définis. Ces éléments sont soumis à controverse et restent pour l'instant du domaine de la recherche. Dans cette situation, les difficultés de mise au point d'une méthode d'évaluation de l'apprentissage dans ce genre de système apparaissent clairement.

Pour ce paragraphe, nous nous baserons principalement sur l'article de D. Jonassen et R. Grabinger [JONAS90], ainsi que sur les colloques "Hypermédias et Apprentissages" de 1991 [PASSAR91] et 1993 [BARON93].

3.1. Champs d'application de l'enseignement hypermédia

Les techniques d'hypertexte et d'hypermédia n'avaient pas, au départ, été conçues dans une optique d'apprentissage. En effet, il s'agissait plutôt de systèmes permettant de consulter rapidement de la documentation et faisaient davantage penser alors à une bibliothèque électronique qu'à un professeur enseignant à des élèves [DEPOV93].

Au cours de ces dernières années, cependant, s'est développée une nouvelle vision du rôle de l'E.A.O. La philosophie sous-jacente est de fournir à l'étudiant un environnement riche, stimulant, où il pourra explorer de grands espaces de connaissances de façon relationnelle, annoter les informations disponibles afin de les rendre plus personnellement significatives et enfin communiquer aux autres étudiants ses remarques sur le cours,... [RHEA91].

Il est apparu que l'hypermédia, conçu dans une simple optique de documentation, était cependant parfaitement adapté à trois formes d'apprentissage :

- la recherche d'informations,
- l'acquisition de connaissances de type expertise,
- la résolution de problèmes.

Pour les autres types d'enseignement, les apports de l'hypermédia seraient plutôt controversés. Par exemple, il a été montré qu'un guidage serré de l'apprenant est essentiel lors d'une acquisition de concepts nouveaux [DEPOV93].

3.1.1. La recherche d'informations

Lorsque l'on apprend, il est essentiel d'avoir accès à de l'information sur la matière à étudier. Un des gros avantages de l'hypermédia par rapport aux systèmes classiques est de mettre à disposition de l'étudiant de nombreuses informations sous diverses formes et présentations.

3.1.2. L'acquisition de connaissances du type expertise

D'après D. Jonassen, les connaissances d'une personne peuvent être représentées par un réseau sémantique constitué d'idées, de relations entre ces idées et des attributs qui les caractérisent. Lorsqu'il y a apprentissage, l'étudiant doit ajouter de nouvelles connaissances à son réseau d'idées. Ensuite, il réorganise ses connaissances pour que son schéma sémantique soit cohérent. Enfin, il optimise ce schéma en fonction du contexte, de la tâche demandée,...

L'acquisition de connaissances n'est cependant pas toujours évidente en hypermédia. En effet, la charge cognitive pour l'étudiant est importante dans ce type d'enseignement [TRICO93]. Non seulement l'élève doit assimiler le contenu des modules (lecture), mais il doit encore comprendre les liens qui existent entre les informations présentées. C'est pourquoi une acquisition efficace de connaissances nouvelles n'est pas véritablement favorisée par l'hypermédia. En revanche, lorsque l'étudiant est déjà bien familiarisé avec la matière, c'est-à-

dire lorsqu'il possède déjà des schémas de pensées sur le sujet, le browsing lui permet de se "mettre au diapason" et d'acquérir par là de nouvelles connaissances. [MAYES93]

3.1.3. La résolution de problèmes

Résoudre un problème est un processus qui se déroule en trois phases. Tout d'abord, l'étudiant doit se "représenter le problème de façon significative" [JONAS90, p.10] c'est-à-dire que le problème doit être exprimé de façon à ce que l'étudiant puisse le comprendre aisément. Or, en hypermédia, on peut avoir accès à différentes représentations du problème. Cela augmente du même coup la probabilité de comprendre celui-ci.

Après s'être représenté le problème, l'étudiant fait appel à ses connaissances pour l'aider à le résoudre. Il applique des heuristiques de recherche sur base de ces connaissances, telles le brainstorming, le raisonnement analogique,... L'hypermédia peut faciliter cette étape, puisqu'elle peut fournir des "représentations d'informations alternatives, tels les liens analogiques" [JONAS90, p.10] et grâce aux systèmes collaborants, favoriser le brainstorming.

La dernière étape en résolution de problème consiste à évaluer les solutions trouvées lors de l'étape 2. Si elles ne sont pas jugées satisfaisantes, le processus reprend à l'étape 2, avec l'espoir de trouver de meilleures solutions. L'hypermédia peut également avoir un rôle à jouer dans cette étape via les environnements collaboratifs.

3.2. Influences sur les élèves

D'après G. Marchionini, on peut regrouper en trois catégories les aspects de l'hypermédia susceptibles d'apporter une modification par rapport à la vision traditionnelle de l'E.A.O. pour ce qui a trait aux élèves :

- l'accès, c'est-à-dire la possibilité de consulter un grand nombre d'information,
- le contrôle, c'est-à-dire la liberté offerte à l'utilisateur,
- la collaboration, c'est-à-dire la facilité d'interaction entre les utilisateurs et la machine ou entre les utilisateurs.

Les bénéfices à retirer pour les étudiants de cette nouvelle forme d'E.A.O. ne vont cependant pas de soi. Au contraire, ce support, s'il est utilisé sans que les étudiants n'aient acquis certaines habiletés, constituerait plutôt une source de difficulté. Prenons tout d'abord le cas de l'accès.

3.2.1. L'accès

L'hypermédia permet de mettre à disposition de l'étudiant d'énormes quantités d'informations sous une grande variété de formats. La première forme d'apprentissage favorisé par l'hypermédia étant la recherche d'informations, il est clair qu'avec ce genre de média, les besoins des étudiants dans ce domaine peuvent être rencontrés. Encore faut-il que les étudiants se retrouvent dans cet espace d'informations interconnectées. Cela nécessite des capacités de recherche de l'information que ne possèdent pas toujours les élèves. Des solutions à ce type de problèmes peuvent être envisagées en fournissant une aide à l'utilisateur sous forme de guides, d'index et de backtracks afin qu'il puisse naviguer efficacement dans l'espace des informations et correctement se représenter la structure de cet espace [BARON93]. On peut également doter l'environnement interactif d'aides intelligentes à l'utilisateur pour conseiller celui-ci dans ses déplacements dans l'hyperdocument [TRICO93]. Une fois cette première difficulté de désorientation passée, constate Georges P. Landow, l'hypermédia est très profitable aux étudiants, "améliorant leur capacité à penser de façon relationnelle [...], les habituant à faire leur propres relations [...], améliorant les discussions de classe, les habitudes de lecture et la capacité à traiter de grandes quantités d'informations" [LANDO90, p.40].

3.2.2. Le contrôle

Les systèmes hypermédiés sont supposés laisser l'utilisateur browser dans l'hyperdocument comme bon lui semble. Tout comme l'accès, cependant, le contrôle peut être source de difficultés supplémentaires pour l'élève. Il existe en effet le risque de voir l'utilisateur perdu dans le document. Il y a ici "une tension d'ordre pédagogique dans le fait de fournir à l'étudiant un médium qui lui permette de choisir comment utiliser ce médium puisqu'ils ne sont plus obligés d'apprendre ce que l'enseignant attend d'eux" [MARCH90, p. 357]. Le problème est encore exacerbé par le fait que rien ne prouve que l'étudiant soit capable de parcourir

l'espace de connaissances de façon pertinente. On considère en effet un peu trop vite "la navigation dans un hypermédia adaptée au consultant puisqu'il est libre de naviguer comme il veut" [TRICO93]. Des solutions pour atténuer ce genre de problème existent. Il s'agit la plupart du temps de limiter le contrôle de l'utilisateur, tout en essayant de ne pas retomber dans un enseignement trop dirigiste. Nous reviendrons à ces systèmes mixtes au cours du paragraphe 3.3.3.

3.2.3. *La collaboration*

Enfin, l'hypermédia facilite les interactions avec la machine et avec d'autres humains. Mais l'interactivité n'est pas le propre des logiciels hypermédiés. Au contraire, nous avons vu au chapitre 1 que c'était une caractéristique fondamentale de tous les systèmes d'E.A.O. Une grande nouveauté cependant réside dans la capacité de l'hypermédia de permettre aux étudiants de créer leurs propres modules et liens. L'hypermédia permet donc une double collaboration, d'une part avec le système, puisque l'étudiant interagit avec celui-ci en le modifiant, et avec les autres étudiants ou le professeur puisqu'il peut faire partager à ceux-ci ces modifications. Bien sûr, ce type de travail de groupe ne "convient pas à tous les enseignements et pose le problème de l'intégrité des informations dans un document hypermédia qui évolue" [MARCH90, p.357]

3.3. Influence sur les concepteurs

L'enseignement hypermédia n'offre pas des défis et des opportunités aux élèves seuls. Nous nous baserons principalement sur l'article de G. Marchionini pour expliquer cet état de fait [MARCH90]. Les concepteurs peuvent également y gagner, aussi bien au niveau de la conception et du développement de logiciels qu'au niveau pédagogique.

L'hypermédia est en effet un outil extrêmement efficace pour illustrer les relations qui existent entre des modules représentant divers concepts. Au niveau du développement, le travail des concepteurs est facilité puisque l'hypermédia permet de modifier facilement modules et liens pour les faire partager à d'autres. En outre, il existe des outils de traitement de texte et de conception graphique, qui peuvent aider le concepteur à écrire ses hyperdocuments. Le principe de collaboration en hypermédia permet en outre de reprendre des modules d'un

didacticiel, de les modifier et de les intégrer dans un autre didacticiel. Pour finir, l'hypermédia pouvant facilement intégrer de l'image et du son, les concepts présentés dans un didacticiel peuvent être illustrés de cette manière.

Les concepteurs connaissent cependant de nouvelles difficultés dues à l'utilisation de systèmes hypermédiés. Ces difficultés proviennent de 3 sources :

- les changements organisationnels,
- la nouveauté du support,
- la tension philosophique découlant de la liberté de l'utilisateur.

3.3.1. Les changements organisationnels

Tout d'abord, une difficulté importante réside dans l'adaptation aux changements organisationnels induits par l'hypermédia en E.A.O. L'I.S.D., très populaire, nécessitait une organisation très centralisée en termes de temps et de gens impliqués [LOGAN82]. En hypermédia, une telle organisation ne semble plus aussi nécessaire, vu les facilités offertes par la collaboration. Cependant, une concertation est toujours nécessaire en ce qui concerne des points clés de l'interface, par exemple, "des décisions doivent être prises en ce qui concerne la taille des fragments de textes, du nombre de liens à placer par écran,..." [MARCH90, p.358]

3.3.2. La nouveauté du support

L'hypermédia étant un instrument relativement récent, il est très difficile pour un professeur de juger l'utilisation d'un cours hypermédia par les étudiants, la littérature étant très dispersée à ce sujet et lui même manquant de recul. De même, des approches prescriptives de développement de didacticiels sont toujours du domaine de la recherche, ce qui n'aide en rien les concepteurs.

3.3.3. La tension philosophique

Une autre source de difficultés dont nous avons déjà parlé plus haut réside dans la tension qui existe entre la philosophie E.A.O. et la philosophie hypermédia. Nous nous baserons pour examiner cet argument sur l'article de T. Mayes, M. Kibby et T. Anderson "Learning About Learning From Hypertext" [MAYES90]. Selon la vue traditionnelle de l'E.A.O., l'étudiant est guidé dans l'interface, le système lui présentant de l'information, des exercices,.. dans un ordre prédéfini par les concepteurs. La philosophie dominante en hypermédia, au contraire, tend à laisser les étudiants libres de choisir quel lien suivre et à quel moment. Cette opposition de vue est d'autant plus problématique en E.A.O. que les enseignants sont accoutumés au fait de "disposer les unités d'interaction (les modules en hypermédia) dans un ordre pédagogique optimal" [LECLER77, p.99]. L'ordre dans lequel les informations sont présentées n'est donc pas sans incidence sur la facilité d'apprentissage des étudiants. En hypermédia, il n'y a pas de "techniques pédagogiques pour s'assurer de la fourniture à l'étudiant de la bonne information au bon moment" [MAYES90, p.229]. L'étudiant est donc seul responsable de la façon dont il traverse l'espace de connaissances. L'impact sur les concepteurs est énorme. Ils ne doivent plus penser le didacticiel en terme d'enseignement, mais plutôt en terme d'apprentissage individuel, ce qui marque encore plus le caractère auto-formatif de ce type d'E.A.O.

Dans les faits, de plus en plus d'auteurs plaident pour une certaine mixité entre contrôle et liberté [BARON93]. L'idée est de continuer à pouvoir se déplacer à volonté dans l'hyperdocument mais de fournir conjointement à l'utilisateur des aides pour se déplacer et rechercher des informations dans la base de connaissances. Il est à noter que les didacticiels hypermédiés actuels utilisent pour la plupart cette approche 'tuteur', qui permet de concilier avec plus ou moins de bonheur la liberté dans la recherche d'informations et le contrôle pédagogique.

3.4. Impact sur les méthodes d'évaluation

Le développement de l'E.A.O. au cours de ces dernières années s'est basé sur la philosophie de l'I.S.D. Malgré une certaine évolution au sein de ce courant, le principe de base demeure, c'est-à-dire que le produit est conçu itérativement, des modifications y étant

apportées sur base des performances des étudiants, les performances à atteindre étant fixées très tôt dans le cycle de développement.

On peut observer cependant que, dans cet optique, on s'intéresse très fortement aux produits de l'apprentissage. Ces produits seront utilisés pour mesurer l'efficacité de l'enseignement. Un cours aura atteint son objectif si les performances des étudiants sont semblables à ceux spécifiés avant que les étudiants n'aient suivi le programme d'enseignement. Formuler les objectifs d'apprentissage, qui serviront d'étalon pour vérifier la qualité du didacticiel est donc une étape essentielle en I.S.D., d'autant plus que c'est sur base de ces mesures que le cours sera modifié.

Mais cette vision itérative basée sur les outputs et leur définition, très populaire pour l'instant et très employée pour tout ce qui touche à l'E.A.O. n'est pas sans poser problème dans le cas de l'enseignement hypermédia.

En hypermédia, il devient très difficile de comparer les performances des étudiants avec des objectifs prédéfinis, et ce pour 3 raisons :

- la mise à disposition des étudiants d'un environnement riche,
- le manque d'expérience des concepteurs,
- le peu de littérature de référence.

3.4.1. La richesse de l'environnement

En hypermédia, la tendance est de fournir des "environnements riches dans lesquels les étudiants peuvent construire leur propre connaissance" [MARCH90, p.359]. L'idée sous-jacente à un apprentissage de ce type est de permettre aux étudiants d'effectuer leur propre évaluation en décidant sur quoi ils veulent être interrogés. Évidemment, cette philosophie peut sembler en contradiction avec l'I.S.D. L'évaluation ne peut plus se baser sur des objectifs prédéfinis si les étudiants peuvent choisir eux-mêmes leurs propres objectifs d'apprentissage. Dans ce cas, il faudrait effectuer l'évaluation de chaque étudiant sur base de ses objectifs personnels.

Mais ainsi que nous l'avons déjà mentionné, la tendance actuelle est de mélanger contrôle et liberté dans ces environnements riches. Cela signifie que des objectifs peuvent être prédéterminés, mais que l'étudiant est libre dans la manière de remplir ces objectifs [TRICO93]. Cette vision va sans nul doute permettre de diminuer les difficultés évaluatives des didacticiels hypermédias.

3.4.2. Le manque d'expérience des concepteurs

Nous avons vu que les concepteurs n'avaient que peu d'expérience de l'utilisation de l'hypermédia en E.A.O. L'évaluation s'en trouvera affectée puisque les concepteurs auront du mal à déterminer si les performances attendues de la part des étudiants sont raisonnables, bien calibrées,... "Il y a donc un important besoin d'établir des données de base sur les effets de telles applications" [MARCH90, p.359].

3.4.3. Le peu de littérature de référence

Il faut souligner un dernier point, c'est que dans la littérature ne se trouvent pas encore clairement expliqués des conseils pour aider les concepteurs à faire des choix sur "les degrés d'accès, de contrôle et de collaboration à affecter à leurs produits" [MARCH90, p.359]. Or, la liberté et le contrôle relatifs de l'étudiant sont des points importants, ayant une influence sur ses performances, comme il a été expliqué au paragraphe 3.2. Difficile donc de concevoir des objectifs si l'on n'a pas une idée de l'influence des décisions prises sur l'accès, le contrôle et la collaboration

Enfin, il est important de ne pas se focaliser uniquement sur les produits de l'apprentissage. Dans le cas de systèmes hautement interactifs comme l'hypermédia, il est également intéressant de prêter attention au processus d'apprentissage lui-même, ainsi qu'à la qualité d'interaction entre l'étudiant et le système.

Cette optique est très nouvelle par rapport à l'évaluation classique dans l'I.S.D. Mais comment faire alors pour mettre au point une évaluation qui se révèle satisfaisante dans le cas

particulier de l'enseignement hypermédia, même si celle-ci se base sur les objectifs personnels des utilisateurs? Nous nous efforcerons d'apporter une réponse à cette question lors des chapitres suivants, en particulier au chapitre 6.

Chapitre 3 :

Apports à l'E.A.O. des philosophies de développement de logiciels

L'E.A.O. est au centre de trois disciplines : la pédagogie, l'informatique et la psychologie cognitive. En tant que telle, elle peut subir des influences en provenance de ces trois sources. Or, ces dernières années, de profonds changements sont intervenus en génie logiciel. En effet, on a constaté dans les méthodologies de développement de logiciels un souci croissant de l'utilisateur et de l'ergonomie de l'interface. Cet état de fait s'exprime à travers deux considérations sur lesquelles les auteurs insistent de plus en plus :

- le centrage sur les utilisateurs et leurs tâches lors de la conception,
- la reconnaissance des mérites de l'évaluation continue.

1. Le centrage sur les utilisateurs et leurs tâches

L'utilisateur devient le centre naturel vers lequel tout le processus de conception doit être tourné. Ce centrage sur l'utilisateur et les tâches qu'il effectue commence très tôt dans le cycle de développement et comprend l'application de différents principes :

- la connaissance de l'utilisateur
- la conception participative

1.1. Connaître l'utilisateur

Nous nous baserons pour ce paragraphe sur l'ouvrage de J. Nielsen "Usability Engineering" [NIELS93].

Connaître l'utilisateur est la première étape de toute méthodologie de développement qui se respecte. Elle consiste pour l'équipe de projet à identifier les futurs utilisateurs du produit que l'on compte développer et à cerner les tâches que ces derniers effectueront à l'aide du logiciel considéré.

1.1.1. Définition

Connaître l'utilisateur revient à étudier les 4 paramètres suivants [NIELS93] :

- les caractéristiques individuelles de l'utilisateur,
- les tâches présentes et désirées,
- l'analyse fonctionnelle des tâches,
- l'évolution de l'utilisateur.

a.) Les caractéristiques individuelles de l'utilisateur

Il s'agit de connaître l'âge, le niveau de formation, l'expérience,...des futurs utilisateur afin de pouvoir "anticiper leur capacité d'apprentissage" [NIELS93, p.74]. La façon dont l'interface sera construite devra évidemment dépendre de ces caractéristiques.

Un autre facteur à prendre en compte à ce niveau est l'environnement de travail dans lequel évoluent les futurs utilisateurs du système. En effet, la manière dont se comportera l'interface doit tenir compte de la façon dont l'utilisateur travaille. Par exemple, si celui-ci est constamment interrompu dans sa tâche, il faut faire en sorte que lorsqu'il y revienne, les données déjà rentrées ne soient pas perdues.

b.) Les tâches présentes et désirées de l'utilisateur

Dans cette étape sont étudiés tous les inputs dont l'utilisateur a besoin pour mener les différentes tâches à bien, ainsi que tous les outputs qui résulteront de l'accomplissement des tâches. Les exceptions dans le traitement des tâches sont aussi à prendre en compte à ce niveau. Mais cette étape ne s'arrête pas là. Elle comprend également la description de ce que l'utilisateur voudrait pouvoir accomplir avec le système.

c.) Analyse fonctionnelle

L'analyse fonctionnelle s'intéresse au pourquoi et au comment de l'exécution des différentes tâches, c'est-à-dire qu'elle décrit la manière dont les tâches sont effectuées et leur raison d'être.

d.) Evolution de l'utilisateur

Les utilisateurs vont évoluer tandis qu'ils interagissent avec le système. Au plus ils l'utiliseront, au mieux ils finiront par le connaître. Il faudra prendre en compte le fait que les utilisateurs vont changer, et donc faire en sorte que le produit soit suffisamment souple pour permettre une bonne utilisation par des novices comme par des experts.

1.1.2. Impacts possibles sur l'E.A.O.

Il faut tout d'abord remarquer que ce souci de mieux connaître les utilisateurs finaux du produit est déjà fort appliquée en E.A.O., où l'on analyse la population cible en termes de connaissances préalables. Ce qui est par contre résolument nouveau, c'est que l'on se penche sur ce que l'utilisateur souhaite pouvoir accomplir avec le système. Cette notion est tout-à-fait absente de l'I.S.D. En général, l'équipe de projet se penche plus sur ce que le professeur désirerait enseigner que sur ce que l'étudiant désirerait apprendre, reléguant au rang de vœux pieux le désir d'une pédagogie centrée sur l'élève. Cependant, des progrès existent en ce sens. Le développement de l'utilisation de l'hypermédia en E.A.O., qui implique en général un enseignement moins dirigiste, basé sur la liberté de glaner des renseignements au gré de déplacements dans l'interface devrait favoriser cette tendance.

1.2. La conception participative

Nous nous baserons pour ce paragraphe sur l'ouvrage de J. Rubin "Handbook of Usability Testing : How to plan, design and conduct effective tests" [RUBIN94].

1.2.1. Définition

Il s'agit de faire participer les utilisateurs au développement du produit, par exemple en adjoignant à l'équipe de projet un de leurs représentants. Derrière cette idée se cache le souci de l'équipe de projet d'avoir constamment sous la main un utilisateur, aussi appelé expert de la

tâche, pour confirmer ou non les idées qu'ils se font sur la façon de penser, le travail,.. des utilisateurs finaux. Pour Rubin, cette idée est de loin préférable à des contacts directs et désordonnés entre les utilisateurs finaux et l'équipe de projet. En effet, des rencontres de ce genre, si elles sont mal structurées (ce qui arrive, d'après Rubin, assez souvent) risquent de déboucher sur "une myriade d'informations contradictoires" [RUBIN94, p.12].

1.2.2. Impacts possibles sur l'E.A.O.

Dans le cadre de développement d'un didacticiel, on se centre sur les étudiants, leur niveau,.. L'idée d'inclure dans l'équipe de projet un étudiant est assez nouvelle, mais elle semble intéressante. En effet, alors que les étudiants seront les utilisateurs du produit, ils sont rarement consultés pour autre chose que d'effectuer des tests. Bien sûr, le contenu des leçons est du ressort des pédagogues, mais il serait néanmoins instructif de demander à un élève comment il perçoit le sujet, quelle représentation de celui-ci lui paraîtrait la plus claire...

2. La reconnaissance des mérites de l'évaluation continue

De plus en plus d'auteurs se penchent sur l'intérêt d'intégrer lors de la conception de logiciels plusieurs phases d'évaluation, dans le but d'améliorer l'ergonomie de l'ensemble du produit. Auparavant, les évaluations intervenaient surtout à postériori, lorsqu'une grande partie du logiciel avait déjà été écrite. L'approche de l'évaluation était alors surtout une approche de diagnostic. Le produit, difficile à utiliser, avait déjà été créé et les modifications possibles à y apporter ne pouvaient dès lors qu'être mineures [KEYSE89].

2.1. Avenir du concept d'évaluation continue

Ces dernières années, de plus en plus d'auteurs préconisent une démarche itérative, intégrant des feedbacks des utilisateurs tout au long du cycle de développement du produit. Les évaluations à ce niveau ont évidemment l'avantage d'être correctives et donc de rendre le produit sensiblement plus ergonomique, puisqu'elles interviennent quand l'investissement n'est

pas encore réalisé et permettent par là des changements importants. Cette démarche semble vouée à un bel avenir, et ce pour 3 raisons, deux d'ordre commercial :

- les économies qui peuvent être réalisées,
- le souci de l'image de marque,

Et l'autre d'ordre plus politique:

- le soutien de la littérature et des clients.

2.1.1. Les économies possibles

En réalisant des évaluations tôt dans le cycle de développement et en modifiant l'idée initiale suite aux résultats de ces évaluations, on peut très vite s'apercevoir des erreurs de conception et y remédier endéans des coûts raisonnables.

De plus, le gain pour les clients sera important. Si le logiciel est conçu en interne, ces bénéfices seront directement visibles. Un produit bien conçu, facile à utiliser et à apprendre, a beaucoup de chance d'augmenter la productivité des utilisateurs. A l'autre extrême, un logiciel mal pensé, peu ergonomique, risque de mener à des performances catastrophiques. Si par contre le logiciel est conçu pour des clients extérieurs, sa difficulté à être utilisé et appris n'incitera pas les clients à revenir.

2.1.2. Le souci de l'image de marque

La qualité des logiciels développés est cruciale pour l'image de marque d'une entreprise de software. L'IHM faisant partie intégrante du produit, il est également important de se préoccuper de sa qualité. C'est en fait devenu un moyen efficace de différencier un produit dans un marché très concurrentiel. "Pour une entreprise de fabrication de software, la facilité d'utilisation de chacun de ses produits contribue à sa réputation générale de fournisseur de qualité" [NIELS93, p.72].

2.1.3. Le soutien de la littérature et des clients

Parmis les auteurs préconisant l'évaluation continue, citons :

"Le feedback quant à la qualité de l'IHM intégré tôt dans le processus de conception est un point critique pour pouvoir effectuer les changements désirés quand ils sont encore possibles."

J. Bradford [BRADF94, p.235]

"C'est un truisme maintenant que les évaluations ergonomiques précoces et un cycle de conception/test/reconception/retest sont des composants importants d'un programme ergonomique".

R. Bias [BIAS94, p.72]

"On a beaucoup parlé de l'importance de la conception itérative[.]. La véritable conception itérative permet une complète révision et un re-pensage de la conception grâce à testage précoce des modèles conceptuels et des idées. Si les concepteurs ne sont pas prêts à faire le saut, alors l'influence de la conception itérative ne peut être que minimale et cosmétique".

J. Rubin [RUBIN94, p.12]

L'idée du contrôle permanent de la qualité des IHM a donc conquis du terrain au cours de ces dernières années. En insistant sur l'importance de l'évaluation continue, les auteurs ont aidé les clients à prendre conscience du fait qu'ils pouvaient exiger de leurs fournisseurs une interface adaptée, facile à utiliser et à apprendre. Des standards internationaux ont vu le jour, certains "ayant même acquis force de loi dans certains pays" [NIELS93, p.9]. L'évolution vers la qualité des interfaces et son pré-requis, l'évaluation continue, semble donc devenir inévitable.

2.2. Les moyens de la mise en oeuvre de cette vision

Nous avons vu qu'au cours de ces dernières années, l'accent est de plus en plus mis sur une évaluation précoce des logiciels. Se pose alors la question de comment intégrer précisément dans le développement d'un logiciel ces étapes d'évaluation ergonomique des interfaces.

2.2.1. Intégration dans le cycle de développement du produit

D'après Grislin, les évaluations devraient porter sur l'ensemble des étapes du développement d'un produit. Le schéma ci-dessous illustre cette idée de développement en 'V'.

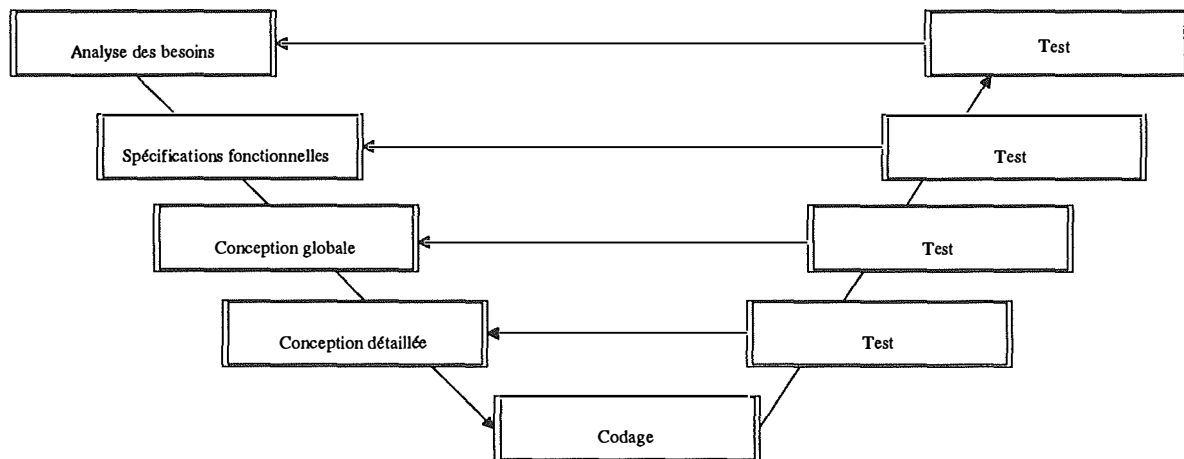


Schéma 4 : Développement en 'V' [GRISL96, p.272]

Il existe de nombreuses méthodes d'évaluation, et on peut se demander si toutes s'appliquent aux mêmes endroits de ce cycle de développement. Nous allons, au cours des chapitres suivants examiner en détail ces méthodes, et les endroits où elles peuvent intervenir précisément.

2.2.2. Classification des méthodes d'évaluation

Il y en a de multiples dans la littérature. Globalement, selon Robert L.Mack et Jakob Nielsen [MACK94], on peut classer ces méthodes en quatre grandes familles :

- les évaluations automatiques, qui tiennent de l'intelligence artificielle et consistent à faire juger l'ergonomie de l'interface par ordinateur ou à faire produire automatiquement une interface simplement en entrant les spécifications de celle-ci.
- les évaluations empiriques, qui mettent en présence des utilisateurs testant le système;
- les évaluations formelles, qui renseignent sur les besoins des utilisateurs en termes d'informations, de fonctions à apporter au système,...

- les évaluations informelles (ou d'inspection), basées sur l'expertise d'évaluateurs spécialisés.

2.3 Impacts sur l'E.A.O.

Tout d'abord, précisons qu'une évolution un peu semblable a eu lieu en E.A.O. Les auteurs soulignent en effet de plus en plus l'intérêt d'appliquer des méthodologies intégrant des feedbacks tout au long du développement. Il ne serait donc pas absurde à priori de penser appliquer des méthodologies de développement assez proches du schéma en V décrit pour les logiciels (cfr. schéma 4), caractérisées par de nombreuses évaluations lors du cycle de développement. Nous verrons plus en détail au chapitre VII comment réaliser cette vision d'évaluation continue dans le cadre de la mise au point d'un didacticiel. Néanmoins, il faut souligner dès à présent que, traditionnellement, lorsqu'on parle d'évaluation dans une perspective logicielle et d'évaluation dans une perspective didacticielle, il existe une différence de taille.

La littérature consacrée à des méthodologies de développement de logiciels et celle consacrée aux méthodologies de développement de didacticiels montre une nette différence de préoccupation quant à l'évaluation du produit. Il faut donc remarquer que, lorsqu'on parle d'évaluation de didacticiels ou d'évaluation de logiciels, on fait référence à deux choses assez éloignées.

En général, les auteurs tournés vers la conception de didacticiels prêtent dans l'évaluation du produit une attention particulière aux performances des étudiants. L'évaluation consiste à comparer les performances effectives des étudiants aux performances attendues et définies au cours des toutes premières étapes du développement, conformément à la philosophie I.S.D. Cela ne signifie nullement que les auteurs ne s'intéressent pas à l'ergonomie de l'interface. Au contraire, il y a depuis quelques années des tentatives visant à donner des recommandations et des conseils quant à la façon dont l'interface doit être construite. Cependant, l'évaluation ergonomique du produit est plutôt laissée dans l'ombre par rapport à l'évaluation pédagogique des performances.

Les auteurs tournés vers la conception de logiciels (au sens strict) sont en général davantage préoccupés par l'ergonomie du produit. Ils intègrent dans le développement différentes étapes d'évaluation de l'ergonomie.

Un apport possible du génie logiciel à l'E.A.O. pourrait constituer en l'élargissement de la notion d'évaluation de didacticiels, de sorte que dans le processus de développement, on prête attention non seulement aux objectifs et à l'évaluation pédagogique de ceux-ci, mais aussi à l'ergonomie même de l'interface.

De plus, une tendance des méthodologies de développement de didacticiels consiste à n'effectuer de véritables tests avec les utilisateurs que fort tard, malgré une évolution certaine au cours des dernières années. Les feed-backs dont nous avons parlé plus haut constituent davantage des approbations de la direction que la prise en compte de l'avis des utilisateurs. L'apport du génie logiciel à ce problème serait d'intégrer une approche d'évaluation continue dans la mise au point des logiciels éducatifs. Ainsi que nous l'avons dit, nous examinerons au cours du chapitre VII comment intégrer ces étapes d'évaluation dans le cycle de développement.

Chapitre 4 :

Les évaluations empiriques

Ce chapitre est en grande partie basé sur l'ouvrage de J. Rubin "Handbook of Usability Testing : How to plan, design and conduct effective test"

Les évaluations empiriques mettent en présence des utilisateurs testant le système. La situation idéale reviendrait à faire effectuer des tests par tous les utilisateurs sur l'ensemble de l'application [NIELS93]. Ce n'est évidemment pas possible dans le cadre de développement d'un logiciel. C'est pourquoi les résultats de ces évaluations ne peuvent être qu'approximés. Rubin distingue trois causes à cet état de fait [RUBIN94] :

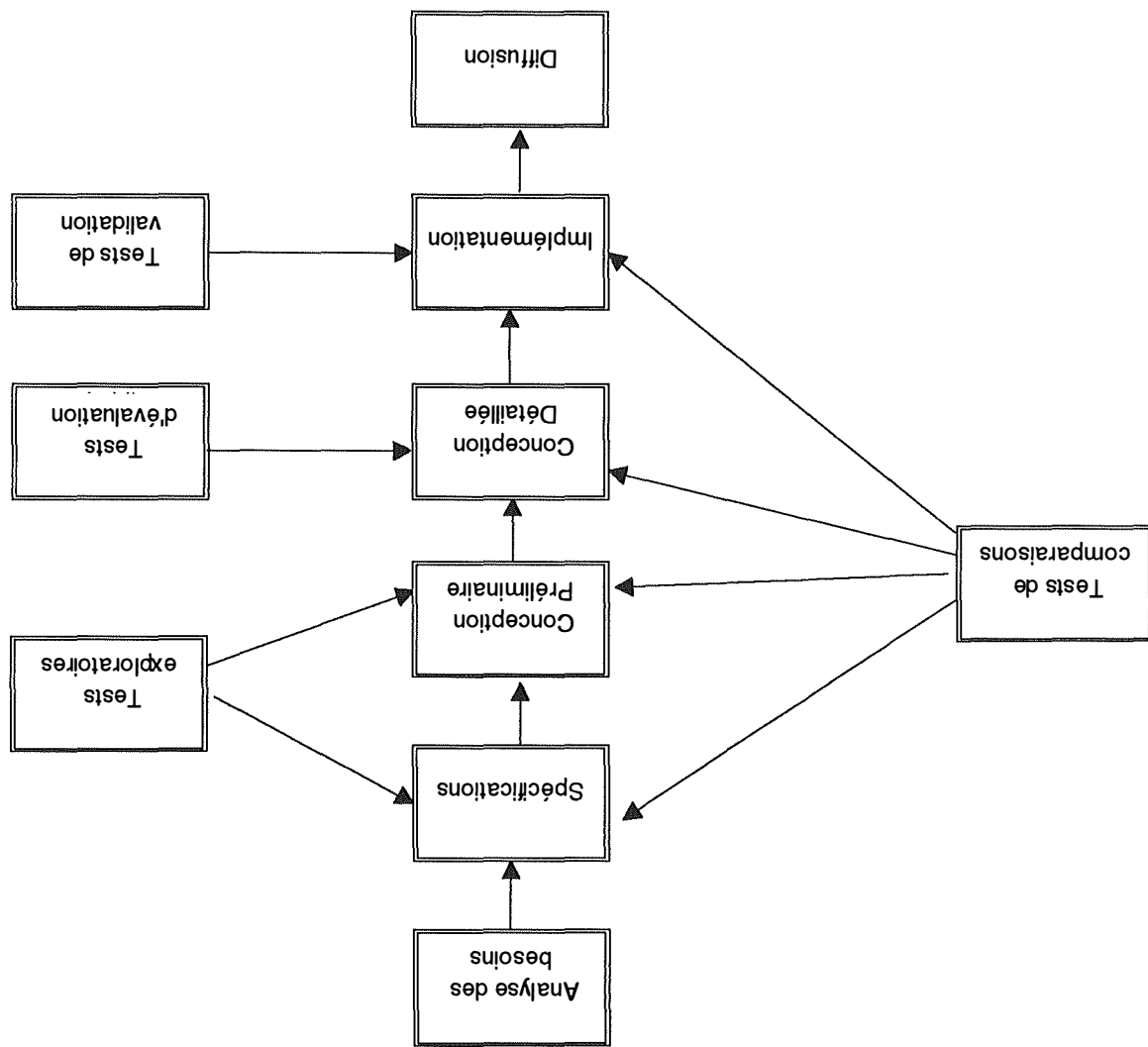
- les tests ne sont pas des indicateurs infallibles de l'ergonomie d'un produit. En effet, le fait que l'on n'ait pas découvert telle ou telle erreur ergonomique ne signifie pas qu'elle n'existe pas et ne sera pas rencontrée par les utilisateurs finaux. La découverte d'erreurs dépend beaucoup de la façon dont les tests ont été conduits.
- les utilisateurs choisis pour réaliser le test ne sont pas toujours exactement représentatifs de l'ensemble de la population des utilisateurs futurs.
- les tests sont toujours artificiels : les utilisateurs risquent de réagir différemment lorsqu'ils se trouvent en situation d'être testés. Leur comportement peut dans ce cas ne pas être tout-à-fait identique à celui montré en situation de travail courante.

1. Les différents types de tests

Les méthodes d'évaluation empirique, décrites par Jeffrey Rubin dans son ouvrage "Handbook of usability testing : how to plan, design and conduct effective tests" [RUBIN94], sont au nombre de quatre :

- les tests exploratoires,
- les tests d'évaluation,
- les tests de validation,
- les tests de comparaison.

Ces tests ne sont pas concurrents, c'est-à-dire qu'ils interviennent chacun à un endroit bien précis dans le cycle de développement du produit, comme indiqué dans le schéma ci-dessous :



Schema 5 : Cycle de développement du produit [RUBIN94, p.32]

1.1. Les tests exploratoires

Les tests exploratoires sont des tests qui ont lieu très tôt dans le cycle de développement du produit, au niveau des spécifications et de la conception préliminaire. Ils ont pour but de donner à l'équipe de projet une meilleure compréhension du modèle mental des utilisateurs. Ils vont "au-delà du produit, puisqu'ils permettent de vérifier les hypothèses sur les utilisateurs" [RUBIN94, p.33]. Ces tests mettent généralement en scène un utilisateur,

confronté à un prototype, mais il est tout-à-fait possible de n'utiliser que des dessins d'écrans ou même une version complète du logiciel existant que l'on souhaite améliorer. Ces tests sont réalisés sous la direction d'un 'moniteur'. Celui-ci discutera de manière assez informelle avec l'utilisateur et lui demandera de 'penser tout haut' quand il explorera le prototype ou réalisera une tâche par son intermédiaire, tout cela dans le but de mieux comprendre ses processus de pensées.

1.2. Les tests d'évaluation

Les tests d'évaluation prennent place assez tôt dans le cycle de développement du produit, juste après que la conception préliminaire ait eu lieu, c'est-à-dire lors de la conception détaillée. Au contraire des tests exploratoires, les tests d'évaluation ne portent plus sur l'utilisateur en tant que tel mais sur la réalisation complète d'une tâche réaliste et sur la découverte d'éventuelles erreurs ergonomiques présentes dans le produit. Des mesures seront récoltées de l'accomplissement de cette tâche, ce qui implique que le moniteur reste plus distant par rapport aux utilisateurs que dans les tests exploratoires. Il ne faut pas en effet influencer les utilisateurs qui effectuent la tâche, puisque c'est la façon dont celle-ci est accomplie qui constitue l'enjeu du test.

1.3. Les tests de validation

Les tests de validation ont lieu tardivement dans le cycle de développement du produit, c'est-à-dire un peu avant sa diffusion, au niveau de l'implémentation proprement dit. Ils poursuivent plusieurs buts. Tout d'abord, ils sont effectués afin de vérifier qu'il ne subsiste pas d'erreurs ergonomiques après les tests d'évaluation. Ensuite, ils ont pour objectif d'établir des comparaisons entre le produit développé et des standards, qui proviennent généralement "d'objectifs ergonomiques développés précédemment dans le projet" [RUBIN94, p.33]. Par rapport aux tests d'évaluation, il ne suffit plus de mesurer des performances. Il faut encore déterminer jusqu'à quel point il faut adhérer aux standards et décider quelles actions entreprendre si le produit ne rencontrait pas ces standards.

1.4. Les tests de comparaison

Les tests de comparaison portent sur l'analyse comparative de plusieurs interfaces, le but étant de déterminer quelle conception est la plus ergonomique, et le pourquoi de cet état de fait. Ces tests de comparaison peuvent être utilisés en "combinaison avec un des trois autres tests" [RUBIN94, p.40].

L'approche de Rubin est très intéressante et novatrice de ce point de vue. En général, les auteurs associent ces tests de comparaison à une approche comparative de différents produits très tôt dans le cycle de développement. Nielsen, par exemple, voit ces tests comme une façon de se forger une opinion sur la manière d'aborder la conception de l'interface [NIELS93]. Or, ces tests ne sont pas, d'après Rubin, associés à un quelconque endroit du cycle de développement du produit. De ce point de vue, la vision de Rubin est plus large que celle de Nielsen.

Au début du cycle de développement, au niveau des spécifications et de la conception préliminaire, ces tests peuvent porter sur différents styles d'interface, pour découvrir quel serait le plus approprié.

Vers le milieu du cycle de développement (conception détaillée), ils peuvent être utilisés pour tester l'ergonomie d'un élément bien précis par rapport à d'autres. Il est à noter que la démarche de Nielsen s'inscrit à contre-courant de ce dernier point, le jugeant trop gaspilleur d'utilisateurs.

Au terme du cycle de développement, au niveau de l'implémentation, ces tests peuvent être utilisés pour comparer le produit à ses concurrents.

2. Capture des données de l'évaluation

Nous nous baserons sur l'article de M. Grislin et C. Kolski pour dépeindre les différentes méthodes utilisées pour capturer les données représentatives de l'évaluation [GRISL96].

Il existe deux grands types de méthodes pour arriver à ce but :

- les méthodes dites participatives, où il y a un véritable dialogue qui se crée entre les utilisateurs qui testent le logiciel et le moniteur du test.
- les méthodes dites non-intrusives où les utilisateurs sont observés sans dialogue avec le moniteur.

2.1. Les méthodes participatives

Ces méthodes mettent en présence un échantillon des utilisateurs et un moniteur. Ce dernier va discuter avec les utilisateurs pour mieux comprendre leur façon de penser, les difficultés qu'ils rencontrent ou les raisons qui les poussent à trouver tel produit peu agréable.

M. Grislin distingue quatre manières de réaliser ce genre d'évaluation :

- recueil de l'expertise,
- analyse des protocoles,
- analyse des incidents critiques,
- construction d'arbre des causes,

2.1.1. Le recueil de l'expertise

Cette méthode est basée sur l'avis d'utilisateurs expérimentés quant aux tâches qu'ils effectuent avec le système informatique existant. Le moniteur discute avec l'utilisateur et décrit les résultats de l'entretien en langage naturel ou à l'aide de graphique.

Cette méthode est particulièrement appropriée pour réaliser les tests exploratoires, où les concepteurs ont besoin de mieux comprendre le travail de l'utilisateur, les informations dont il a besoin pour le réaliser, et sa manière de réfléchir.

2.1.2. L'analyse des protocoles

L'analyse des protocoles est une méthode issue du recueil de l'expertise. Elle poursuit les mêmes objectifs que cette dernière. L'utilisateur en interaction avec le système pense tout

haut sa tâche. Il est enregistré pendant ce temps par une caméra et un magnétophone. Le moniteur décrit par après les informations pertinentes relevées sur base de ces enregistrements.

Cette technique peut également être utilisée pour mener à bien les tests exploratoires, où Rubin insiste sur la nécessité de faire penser l'utilisateur tout haut pour bien comprendre ses processus mentaux.

2.1.3. L'analyse des incidents critiques

La méthode des incidents critiques se base d'une part sur l'observation des utilisateurs en situation de test et d'autre part sur des entretiens avec ceux-ci. On relève sur ces bases la liste de tous les incidents et dysfonctionnements rencontrés par les utilisateurs.

2.1.4. La construction d'arbre des causes

Il s'agit de décrire graphiquement "un enchaînement causal sous la forme d'une arborescence (avec connecteurs ET et OU) en partant de l'événement indésirable et en cherchant ses causes possibles" [GRISL96, p. 274]. Les problèmes ergonomiques ayant entraîné l'erreur de l'utilisateur sont ainsi clairement mis en évidence.

2.2. Les méthodes non-intrusives

Ces méthodes, contrairement aux méthodes participatives, ne mettent plus forcément en présence un échantillon des utilisateurs et un moniteur. Ou du moins, ce dernier a un rôle beaucoup moins actif. Les utilisateurs sont simplement observés tandis qu'ils interagissent avec le système, sans qu'il n'y ait dialogue avec le moniteur. M. Grislin distingue cinq supports propres à réaliser ce genre d'évaluation :

- l'enregistrement vidéo,
- les traces écrites du travail,
- les questionnaires,
- les mouchards électroniques,

- les tactiques visuelles.

Ces méthodes ne nécessitant que peu d'interactions entre le moniteur et les participants au test, elles semblent toutes indiquées pour réaliser des tests de validation et d'évaluation.

2.2.1. L'enregistrement vidéo

Il s'agit simplement d'utiliser une caméra pour filmer les utilisateurs en situation de test. Avec cette façon de faire, on évite le biais dû à l'intervention du moniteur. Celui-ci en effet n'intervient pas, laissant les utilisateurs travailler seuls devant le logiciel. Après la séance, celui-ci devra néanmoins repasser la vidéo aux utilisateurs et avoir un entretien avec eux pour qu'ils lui donnent leurs commentaires sur le film.

2.2.2. Les traces écrites du travail

L'utilisateur prend note par écrit de tous les problèmes qu'il rencontre et ce pas à pas dans la réalisation d'une tâche. Cette méthode peut être complétée par une étude des notes écrites, documents de conduite et cahiers de doléances de l'utilisateur.

2.2.3. Les questionnaires

Il s'agit pour les utilisateurs de remplir après-coup une liste de questions suite aux tests avec le logiciel. Ces questionnaires peuvent contenir des questions fermées, avec une échelle de valeurs, ou des questions ouvertes, où l'utilisateur peut donner son avis. Il peut cependant y avoir un biais dû au fait que "les questionnaires sont remplis en dehors des situations de travail" [GRISL96, p.274]

2.2.4. Les mouchards électroniques

Toutes les interactions de l'utilisateur avec le système sont enregistrées sur un fichier. Cette méthode suit le déroulement de chacune des actions de l'utilisateur qui accomplit sa tâche. On tente de plus en plus d'associer à ces systèmes de capture des systèmes automatiques d'analyse.

2.2.5. Les tactiques visuelles

Les tactiques visuelles se basent sur l'observation du mouvement des yeux des utilisateurs lorsqu'ils effectuent une tâche à l'aide d'un logiciel. Cette méthode met en évidence les lacunes éventuelles de placement des informations dans les écrans de l'interface.

3. Description des différentes étapes de réalisation d'un test

J. Rubin distingue six étapes dans la réalisation des tests empiriques [RUBIN94]:

- développement du plan de test,
- sélection des participants,
- préparation des matériaux de test,
- conduite du test
- débriefing des participants,
- transformation des données recueillies en découvertes et recommandations.

3.1. Le développement du plan de test

Le développement du plan de test est une étape des plus importantes puisque c'est sur cette base que repose tout le processus d'évaluation. Développer un plan de test est un processus complexe, qui procède en neuf étapes :

- but du test, où l'on rapporte les raisons générales qui poussent à entreprendre une évaluation.

- objectifs du test, où l'on note ce que l'on veut apprendre de l'évaluation. Ces objectifs doivent être décrits le plus précisément et le plus clairement possible.
- profils des utilisateurs, où l'on dégage les caractéristiques des utilisateurs finaux du produit.
- plan de test. Il s'agit de détailler "chaque facette du test depuis le moment où les participants arrivent jusqu'à celui où ils partent" [RUBIN94, p.87].
- la liste des tâches que devront accomplir les utilisateurs durant la session, avec pour chacune de ces tâches, une brève description de celle-ci, les matériaux nécessaires et l'état des machines requis pour son exécution. Il importe également d'ajouter à cette liste une description de la bonne exécution des tâches et le temps limite imparti pour les mener à bien.
- environnement et équipement requis pour mener le test à bien. Il ne s'agit nullement ici de s'attarder sur le matériel de capture de données, mais sur l'équipement à utiliser pour simuler un environnement semblable à celui que les utilisateurs connaissent dans leur travail.
- rôle du moniteur, où l'on spécifie quand et de quelle manière le moniteur du test devra intervenir auprès des utilisateurs.
- données à collecter. Ces données peuvent être orientées performance, auquel cas elles reprennent des mesures collectées en observant les utilisateurs, comme le taux d'erreurs, le temps requis à accomplir une tâche,... Ou bien ces données peuvent représenter les avis, les processus mentaux des utilisateurs, auquel cas on les appelle données de préférence. Le choix du type de données à prendre en considération dépend évidemment des objectifs du test.
- présentation du rapport du test. Il s'agit de spécifier comment communiquer à l'équipe de projet les résultats de l'analyse. On peut, par exemple, choisir de communiquer ces résultats lors d'une session d'information.

3.2. La sélection des participants

Une sélection avisée des participants au test est un critère majeur de validité de l'évaluation. En effet, il est essentiel de sélectionner des personnes qui soient pleinement représentatives des utilisateurs futurs du logiciel. A des personnes fortement représentatives

Sélectionner le nombre de participants revient à faire un choix entre le degré de fiabilité du test et les contraintes de coûts et de temps. Pour obtenir des résultats statistiquement corrects, il faut un nombre suffisant d'utilisateurs pour pouvoir généraliser les résultats de l'évaluation à l'ensemble de la population. En général, cependant, on se contente de la "découverte du plus grand nombre d'erreurs ergonomiques possible dans les meilleurs délais" [RUBIN94, p. 128]. Des études auraient, selon J. Rubin, montré qu'on peut raisonnablement attendre d'un groupe de 5 participants la découverte de près de 80 % des problèmes ergonomiques. Il ne s'agit bien sûr que d'une approximation, et de plus, cela ne signifie pas que les 20 % de problèmes restants ne sont pas des erreurs d'IHM importantes. C'est pourquoi il est conseillé d'augmenter le nombre de participants jusqu'à 12 pour diminuer la probabilité de passer à côté d'erreurs ergonomiques.

3.2.2. Nombre de participants

Une description du profil des utilisateurs doit normalement avoir été menée au cours de l'étape de planification. On range les profils des utilisateurs selon un tableau reprenant d'une part les groupes auxquels les utilisateurs appartiennent et qui vont utiliser le logiciel dans des optiques différentes, par exemple les secrétaires, la direction, ... et d'autre part le niveau d'expérience des personnes composant les groupes. Ce niveau d'expérience doit être exprimé sur base de "point de référence objectif comme le temps passé à utiliser un produit ou la fréquence avec laquelle une tâche spécifique est accomplie" [RUBIN, p.127]. Ce critère a pour avantage de ne pouvoir être mal interprété ou mal compris.

3.2.1. Classement des utilisateurs par catégories

des futurs utilisateurs, il peut être intéressant d'ajouter quelques utilisateurs tout-à-fait novices, même s'ils ne représentent pas un pourcentage important des utilisateurs finaux. S'ils parviennent à se débrouiller pendant le test, c'est une des meilleures garanties que l'interface est ergonomique. S'ils ne s'en sortent pas, cela ne veut pas forcément dire que le produit est mal pensé. Cela révèle simplement "des indices sur comment fixer les problèmes fondamentaux de l'intuition, de l'orientation ou de l'organisation durant le re-design" du logiciel [RUBIN94, p.129].

Si on utilise un tableau des compétences et catégories comme décrit ci-dessus, il convient de choisir des participants représentatifs de toutes les cellules. D'après Rubin, il faut compter 4 ou 5 participants pour chacune de celles-ci.

3.3. La préparation des matériaux de test

Une des activités les plus importantes lorsqu'on désire effectuer un test empirique est de développer soigneusement les matériaux nécessaires qui "seront utilisés pour communiquer avec les participants, collecter les données" [RUBIN94, p.141]. J Rubin distingue 8 catégories principales de matériaux de test :

- le script d'orientation,
- le questionnaire de background,
- les instruments de récolte de données,
- le questionnaire de pré-test,
- les scénarii des tâches,
- les matériaux nécessaires pour donner l'enseignement pré-requis,
- le questionnaire de post-test,
- les sujets de débriefing.

3.3.1. Le script d'orientation

Ce script a pour but d'informer les participants de comment va se dérouler la session, des méthodes de récolte de données employées, et de ce qu'eux-mêmes devront faire. Il est lu à haute voix aux participants par le moniteur.

3.3.2. Le questionnaire de background

Ce questionnaire est rempli par les participants avant que ne débute le test en lui-même. Il a pour but de mieux cerner les capacités, l'expérience et les préférences des participants quant au produit. Il servira au moniteur pour interpréter les comportements et les

performances lors du test. Les informations recueillies là sont plus spécifiques que celles provenant des profils écrits lors du plan de test, où l'on se contentait de classer les utilisateurs en larges catégories.

3.3.3. Les instrument de récolte de données

Il existe bon nombre d'instruments de récolte de données. Nous en avons déjà vu quelques uns au point 2. Les données récoltées peuvent globalement se décompose en données de performance et données de préférence .

Les données de performance sont des "mesures objectives du comportement, tels le taux d'erreurs, le temps nécessaire à l'accomplissement de la tâche, et une addition des éléments de comportements observés" [RUBIN94, p.156]

Les données de préférence sont pour leur part sujettes à l'interprétation puisqu'elles consistent à demander aux participants leurs préférences, opinions,...

Ces deux types de mesures peuvent être à leur tour interprétés de deux façons. Tout d'abord, dans le cas des données de performance, on peut agréger les résultats et ainsi travailler statistiquement. C'est l'interprétation quantitative. On peut aussi analyser ces données de performance de manière qualitative, en exposant les "endroits où les utilisateurs n'ont pas compris le modèle conceptuel du produit" [RUBIN94, p.156].

Dans le cas des données de préférence, on peut les analyser quantitativement en additionnant le nombre de remarques des participants sur telle ou telle portion de l'interface. Ou on peut les interpréter qualitativement en essayant de découvrir quels sont les éléments précis qui gênent les participants dans la portion de l'interface considérée.

3.3.4. Le questionnaire de pré-test

Ce questionnaire poursuit différents objectifs. Tout d'abord, il aide le moniteur à mieux comprendre les attentes des utilisateurs et ce, avant tout testage. Pour se faire, les participants

sont confrontés à un écran de l'interface montré par le moniteur. Celui-ci leur demande alors leurs premières impressions sur ce qu'ils ont vu du produit.

Couplé à un questionnaire de post-test, le pré-test permet de voir l'évolution des opinions à priori des participants.

Le pré-test permet aussi d'avoir une connaissance approfondie des capacités des participants et d'ainsi mieux évaluer les résultats du test.

3.3.5. *Les scénarii des tâches*

Les tâches à effectuer par les participants sont décrites dans le scénario des tâches, qui constitue en quelque sorte une version améliorée, étoffée, de la liste des tâches prévue dans le plan de test. Les utilisateurs devant bien sûr en avoir connaissance, ces scénarii leur sont lus ou distribués sur papier.

Ainsi que le souligne J. Rubin, les scénarii des tâches doivent inclure dans leur description les résultats attendus, les messages que les utilisateurs verront tandis qu'ils exécuteront la tâche et l'état du système lorsqu'ils commenceront à travailler.

3.3.6. *Les matériaux nécessaires pour donner l'enseignement pré-requis*

Il peut être utile de donner aux participants un certain enseignement avant de les lancer dans le testage proprement dit. En général, on donne un enseignement dans les cas suivants :

- l'utilisation du produit nécessite des connaissances particulières et on n'est pas sûr que les utilisateurs les possèdent.
- on souhaite tester l'utilisation d'un produit plus avant dans la courbe d'apprentissage. En effet, il y a des différences de comportements entre un utilisateur novice et un autre plus expérimenté. Leurs commentaires sur le produit varient aussi fortement selon leur niveau. On peut alors être tenté d'inclure des participants plus expérimentés dans le test pour avoir leurs opinions. Mais dans le cas d'un nouveau produit, il n'est pas possible

de trouver ce type de participants. L'enseignement permet alors de faire progresser les utilisateurs sur la courbe d'apprentissage jusqu'au niveau souhaité.

- on ne souhaite tester que certaines parties du produit nécessitant plus d'expérience de la part des utilisateurs.

3.3.7. *Le questionnaire de post-test*

Le but de ce questionnaire, en général écrit, est de récolter un maximum d'information de préférence. Ces questionnaires sont distribués à tous les participants qui les remplissent après avoir effectué le test. Les questions posées aident à éclairer le moniteur sur des points non directement observables, comme les opinions des utilisateurs, leurs préférences,...

3.3.8. *Les sujets de débriefing*

Il s'agit simplement de la liste des sujets à aborder absolument durant la session de débriefing. Nous verrons plus en détail le débriefing au point 3.5.

3.4. La conduite du test

La conduite du test proprement dit se déroule dans une même pièce, avec en général une douzaine de participants travaillant sous l'oeil attentif d'un moniteur. Cette méthode peut être utilisée pour les "quatre types de tests présentés plus haut, la différence principale résidant dans le type d'objectif poursuivi, c'est-à-dire plus conceptuel pour un test exploratoire, plus orienté comportement pour les tests d'exploration et de validation" [RUBIN94, p.213].

Le test en lui-même est une étape fondamentale, qui doit respecter certaines contraintes. La plus importante d'entre elles est la cohérence. Il importe en effet que les tests se déroulent de la même façon peu importe la session. L'ensemble des participants doivent en effet travailler sous les mêmes circonstances et disposer d'un matériel identique. Pour s'assurer de cette cohérence, il est important pour le moniteur d'utiliser des check-lists. Elles

permettront en effet de s'assurer que ce dernier n'oublie pas une activité et que tous les participants exécutent bien la même tâche.

Rubin distingue trois check-lists, ou ensemble de démarches à effectuer, pour aider le moniteur. La première planifie les activités devant avoir lieu \pm 2 semaines avant que ne débutent les tests proprement dits, la seconde se préoccupe de ce qu'il faut faire la veille et la troisième décrit les activités se déroulant le jour même.

3.4.1. La première check-list

Elle décrit 4 activités à mener à bien 2 semaines avant les tests :

- faire le test soi-même, ce qui amènera à le réviser pour qu'il soit tout-à-fait au point au niveau du temps imparti, des questionnaires à remplir,...
- effectuer un test pilote en interne, afin de découvrir les problèmes du processus de test avant la venue des participants. De là, on raffine et révise une nouvelle fois le plan de test.
- réviser le produit sur base des erreurs ergonomiques trouvées à l'étape précédente. En effet, il est inutile tout autant que coûteux de faire le test sur un produit dont on connaît déjà les erreurs.
- vérifier l'équipement et l'environnement du test.

3.4.2. La seconde check-list

Elle décrit 4 activités à mener à bien la veille du test :

- vérifier que l'équipement vidéo est prêt,
- vérifier que les programmes et les ordinateurs fonctionnent bien,
- rassembler tous les documents papiers dont auront besoin les participants comme les questionnaires, les scénarii de test,...
- s'assurer que les participants viendront bien,
- vérifier à nouveau l'équipement et l'environnement du test.

3.4.3. La troisième check-list

Elle décrit les activités à mener le jour du test. Dans la littérature, on distingue en général 8 étapes à enchaîner. Deux ont lieu dans la salle d'attente :

- lire le script d'orientation aux participants. Cette étape a pour but de les familiariser avec les techniques de récolte des données qui seront menées par le moniteur.
- s'assurer que les participants remplissent tous les questionnaires pré-tests, c'est-à-dire ceux qui ont trait au produit. "A moins qu'il ne soit nécessaire pour les participants de voir le produit d'abord (possibilité de donner leurs premières impressions), les questionnaires peuvent être remplis dans la salle d'attente" [RUBIN94, p.230]

Les six autres étapes ont lieu après que les participants soient rentrés dans la salle de test. Il s'agit alors de :

- fournir l'enseignement pré-requis au test si spécifié dans le plan du test,
- distribuer ou lire le scénario des tâches aux participants,
- enregistrer l'heure à laquelle le test commence, observer les participants et collecter les données critiques,
- s'assurer que les participants remplissent tous les questionnaires post-test, et ce avant toute discussion sur le test pour éviter les biais,
- débrieffer les participants. Cette étape sera expliquée au paragraphe suivant (3.5).
- organiser la récolte des données et les feuilles d'observation.

3.5. Le débriefing des participants

C'est lors de cette étape que le moniteur du test va pouvoir éclaircir les problèmes survenus pendant la session. Il s'agit de comprendre le pourquoi des erreurs commises par les participants, quelles sont les difficultés qu'ils ont rencontrées,... Il existe toute une série de techniques permettant de réaliser au mieux cette étape :

- le "replay test",
- la révision des conceptions alternatives,
- la technique du "De quoi vous souvenez-vous?",
- la technique du "Comment auriez-vous fait?",

- la technique de l'avocat du diable.

3.5.1. Le replay test

Cette technique consiste à solliciter la mémoire des participant sur la façon dont s'est déroulé le test. L'utilisation de caméra et de magnétoscope est recommandée pour utiliser ce type de technique, bien qu'il soit possible de ne se baser que sur les notes prises par le moniteur durant le tests.

Le moniteur rafraîchit la mémoire des participant en repassant des morceaux de l'enregistrement vidéo pris au moment du test (ou en relisant tout haut ses propres notes). Cette façon de faire est très utile pour aider les utilisateurs à dire les impressions et les pensées qui les ont traversés durant le test et qu'ils auraient omis de dire sinon.

3.5.2. La révision des conceptions alternatives

Cette technique est seulement utilisable aux toutes premières étapes de la conception du produit, ce qui correspond aux tests exploratoires. A ce moment, il peut être intéressant de se pencher sur les préférences des utilisateurs en leur montrant différentes versions du produit.

3.5.3. La technique du "De quoi vous souvenez-vous?"

Cette technique est utilisée pour tester l'ergonomie de tel ou tel composant précis de l'interface et dont il est important que l'utilisateur remarque la présence. Il est clair que le moniteur ne doit pas demander aux participants s'ils ont bien vu l'élément durant le test. Cela "attirerait l'attention sur cet élément" [RUBIN94, p.250] et le test en serait du même coup biaisé. La technique du "De quoi vous souvenez-vous?" intervient alors.

Durant le débriefing, le moniteur montre aux participant le dessin d'un écran semblable à un de ceux qui compose l'interface originale, mais dont l'élément sur lequel on souhaite des

renseignements est enlevé. Le moniteur demande alors aux participants s'ils se souviennent de quelque chose.

3.5.4. La technique du "Comment auriez-vous fait?"

Cette technique est sujette à controverse dans la littérature. Elle consiste en fait à demander aux utilisateurs, après qu'ils se soient trompés, comment ils auraient construit tel ou tel élément de l'interface afin de faciliter leur tâche. Cette façon de faire peut être utile dans des situations hypothétiques, où l'interface n'est pas encore disponible afin de s'assurer des préférences des utilisateurs. Mais l'utiliser après un test, donc à posteriori, est déconseillé par J. Rubin, car du fait que les participants ont déjà rencontré le problème, leur réponse sera inévitablement biaisée.

3.5.5. La technique de l'avocat du diable

Cette technique est utilisée pour voir jusqu'à quel point les participants sont sûrs de leur position sur un point précis, délicat. Elle est également utilisée dans les cas où "les participants ont rencontré des difficultés durant le test mais n'en soufflent mot lors du débriefing" [RUBIN94, p. 253].

Le moniteur cesse dans cette technique d'adopter une position neutre vis-à-vis des remarques des participants. Il doit au contraire prendre une position tout-à-fait à l'opposé des avis des participants.

3.6. La transformation des données recueillies en découvertes et recommandations

Cette étape consiste à découvrir le pourquoi des erreurs commises par les participants dans la réalisation des tâches prescrites et la façon dont il faudrait améliorer l'interface pour que ces problèmes disparaissent. Elle se décompose en trois sous-étapes principales :

- compilation et résumé des données,
- analyse des données,

- développement de recommandations.

3.6.1. *Compilation et résumé des données*

La compilation des données est un processus qui consiste à "placer toutes les données récoltées dans un format qui permet de voir les correspondances" [RUBIN94, p.259]. Ce processus de récolte et de compilation doit débiter dès la session de test.

Le processus de résumé, lui, se déroule après la compilation. Il peut se rapporter à différents types de données. En général, on considère le résumé de données de performance et de préférence. Ces données sont résumées selon différentes techniques, que nous allons passer en revue. Il peut être intéressant également de répartir ces résultats selon les catégories de participants et/ou selon les différentes versions du produit si le test porte sur la comparaison d'interface existantes.

a.) Le résumé des données de performance

Le résumé des données de performance peut être relatif au temps d'exécution de la tâche. Dans ce cas, on effectue plusieurs calculs.

- le temps moyen d'exécution, qui indique combien de temps il a fallu en moyenne aux participants pour exécuter la tâche.

Temps Moyen = Somme des temps de tous les participants

Nombre de participants

- le temps médian d'exécution, qui est le temps central dans la liste des temps d'exécution triés par ordre ascendant. Contrairement au temps moyen, il n'est plus influencé par des temps d'exécution extrêmement longs ou courts.
- l'écart compare le meilleur temps d'exécution au plus long. De très grosses différences entre les deux est en général révélateur d'une différence de compétences entre les participants au test.
- la déviation standard, ou s , mesure les différences entre les temps d'exécution, le "degré avec lequel ils diffèrent l'un de l'autre" [RUBIN94, p.261].

$$s = \text{SQRT} (\Sigma x^2 / n - (\Sigma x / n)^2),$$

n étant le nombre total de score,
 Σx étant la somme de tous les scores

Le résumé des données de performance peut également être relatif à la bonne exécution de la tâche. On compte alors, par ordre croissant d'inclusion :

- Le pourcentage des participants ayant exécuté correctement la tâche dans le temps imparti.
- Le pourcentage des participants ayant exécuté correctement la tâche peu importe le temps imparti. Les utilisateurs faisant des erreurs mais sont capables de se corriger et d'aboutir finalement à une bonne exécution de la tâche.
- Le pourcentage des participants ayant exécuté correctement la tâche dans le temps imparti, y compris ceux qui ont eu besoin d'assistance. Cette mesure est la plus inclusive des trois. C'est-à-dire qu'un pourcentage relativement faible ici est une indication désastreuse de l'ergonomie du produit.

b.) Le résumé des données de préférence

Ces données sont tirées des enquêtes, des questionnaires de post-test et des sessions de débriefing. Vu le caractère plus subjectif de ce genre de données, il peut être plus difficile de faire un résumé. Pour les questionnaires à choix multiples, c'est relativement simple. Il suffit en effet de compter le nombre de réponses identiques. En ce qui concerne les réponses à des questions ouvertes, le problème est plus compliqué. Il s'agit de "grouper les réponses similaires en des catégories sémantiques" [RUBIN94, p.265]. Enfin, pour les sessions de débriefing, il faut essayer de retirer des interviews les commentaires critiques et de les placer dans telle ou telle catégorie.

3.6.2. L'analyse des données

L'analyse doit tout d'abord porter sur les tâches qui ne répondent pas à certains critères. "Une tâche qui ne rencontre pas les critères est une tâche qu'un pourcentage prédéterminé de participants n'a pas réussi à exécuter correctement" [RUBIN94, p.274]. Une tâche dont 70 % des utilisateurs n'arrivent pas au bout est qualifiée de problématique.

A l'intérieur de ces tâches ne répondant pas aux critères, il faut alors découvrir quelles sont les causes de ces mauvaises performances des utilisateurs, c'est-à-dire les éléments de l'interface qui sont sources d'erreurs. Pour cela, le moniteur se base sur les enregistrements vidéo, ses notes,...

Une fois les sources d'erreurs identifiées, il faut encore assigner un taux critique à ces problèmes. Ce taux indiquera la priorité de résolution des problèmes rencontrés. Pour Rubin, le taux critique est égal à la somme de deux facteurs, déduits de la sévérité du problème et de sa probabilité d'occurrence. La sévérité est estimée selon une échelle de un (produit inutilisable) à quatre (problème cosmétique). Quand à la probabilité d'occurrence du problème, on lui assigne également un taux variant de 1 à 4. Si le problème se produit plus de 90 % du temps d'utilisation, on lui assignera un taux de 4. Si le problème se produit moins de 10 % du temps d'utilisation, on lui assignera par contre un taux de 1. De 11 à 50 % et de 51 à 89 %, les taux seront respectivement de 2 et 3. Une fois la sévérité du problème et sa probabilité d'occurrence traduits en terme de taux, il ne reste plus alors qu'à effectuer l'addition de ces deux résultats pour obtenir le taux critique.

Pour terminer, il peut être intéressant d'effectuer des analyses séparées pour les différents groupes de participants ou les différentes versions du produit.

3.6.3. Le développement de recommandations

Il s'agit de "prendre toutes les informations analysées et de les transformer en recommandations pour agir" [RUBIN94, p.283]. Il s'agit là d'un processus extrêmement difficile, pouvant nécessiter des connaissances en ergonomie, en psychologie cognitive,... C'est pourquoi J. Rubin recommande de développer ces recommandations en équipe pluridisciplinaire, afin de disposer de plusieurs backgrounds de connaissance.

Pour les erreurs purement ergonomiques, c'est-à-dire qui se réfèrent uniquement à la qualité de l'interface, il est conseillé de s'inspirer des nombreuses guidelines présentes dans la littérature pour réduire au maximum le risque de recommandations inappropriées.

Chapitre 5:

Les autres méthodes d'évaluation ergonomique

Le processus d'évaluation via les méthodes empiriques est relativement lourd, et l'investissement nécessaire pour le mener à bien est important. Or, depuis quelques années sont apparues dans la littérature des méthodes d'évaluation ne faisant plus appel à un panel d'utilisateurs finaux. Plus souples, moins coûteuses, ces évaluations sont appelées à jouer un rôle croissant dans le processus global d'évaluation.

1. Les évaluations informelles

Ce type d'évaluation, aussi appelée inspection d'experts, met en présence le produit à tester, un prototype ou même une maquette papier avec un ou plusieurs inspecteurs chargés de découvrir les problèmes ergonomiques présents dans l'interface.

Robert L.Mack et Jakob Nielsen, ont décrit dans leur ouvrage "Usability Inspection Methods" [MACK94] les différentes méthodes d'évaluation informelle. Citons, entre autres :

- les évaluations heuristiques,
- les guidelines,
- les promenades pluridisciplinaires,
- les inspections de cohérence,
- les inspections de standards,
- les promenades cognitives
- les inspections ergonomiques formelles,
- les inspections de caractéristiques.

1.1. Les évaluations heuristiques

Nous nous baserons pour ce paragraphe sur l'article de J. Nielsen "Heuristic Evaluation" [NIELS94] extrait de l'ouvrage "Usability Inspection Methods" [MACK94].

1.1.1 La méthodologie de travail

D'après J. Nielsen, les évaluations heuristiques sont effectuées par plusieurs experts, qui examinent l'interface, jugent de l'ergonomie des éléments du dialogue et recherchent les erreurs grâce à des heuristiques, comme il en existe beaucoup dans la littérature. Les experts se doivent de travailler de façon indépendante dans un premier temps, et ce, pour éviter d'être influencés par les découvertes des autres évaluateurs. Durant la session d'inspection, les évaluateurs sont observés par un moniteur, auquel ils peuvent demander des explications ou de l'aide quant à l'utilisation de l'interface. Il faut compter entre une et deux heures de travail par évaluateur pour mener à bien son inspection.

Ensuite, les évaluateurs mettent en commun leurs résultats lors d'un débriefing. A cette réunion participent également le moniteur et des représentants de l'équipe de projet. Au cours du débriefing, les évaluateurs assignent un taux de sévérité aux erreurs trouvées. Il représente la gravité de l'erreur et sa priorité à être traitée en fonction de différents facteurs :

- le nombre d'occurrences de l'erreur,
- l'impact de l'erreur, ce qui revient à se poser la question de savoir si l'erreur peut gravement hypothéquer le caractère ergonomique de l'interface,
- la persistance de l'erreur, c'est-à-dire si elle revient continuellement ou s'il ne s'agit que d'un problème passager.

Il est à remarquer que cette méthode heuristique aboutira à un output sensiblement identique à celui issu des tests empiriques décrits au chapitre 4, c'est-à-dire à une liste de problèmes avec un taux de sévérité associés. Ces deux méthodes semblent donc concurrentes. Les auteurs cependant, semblent d'accord pour dire que les méthodes heuristiques et empiriques sont plus appelées à se compléter qu'à s'exclure. Nous reviendrons à cette constatation lors de la comparaison entre ces deux approches (chapitre 6).

1.1.2. Nombre et caractéristiques des évaluateurs

La somme du nombre d'erreurs trouvées par tous les évaluateurs dépend de deux facteurs principaux.

Il s'agit de :

- le nombre d'évaluateurs,
- les compétences de ces évaluateurs.

Le nombre moyen d'experts à utiliser pour mener à bien ces évaluations est compris d'après Nielsen entre 3 et 5. Un seul évaluateur trouve de l'ordre de 35 % des erreurs ergonomiques. "Cependant, vu que différents évaluateurs tendent à trouver différents problèmes, il est possible d'obtenir de meilleures performances en agrégeant les découvertes de plusieurs évaluateurs" [NIELS94, p.32].

En effet, le nombre total d'erreurs peut être approximé par la formule suivante :

$$\#erreurs(i) = N(1 - (1 - \lambda)^i)$$

où $\#erreurs(i)$ est le nombre d'erreurs trouvées par les i évaluateurs de l'inspection, où N est le nombre total d'erreurs présentes dans l'interface et où λ est la pourcentage d'erreurs trouvé par un seul évaluateur. Ce paramètre varie de 19 à 51% avec une moyenne de 34%.

Le nombre d'évaluateurs n'est pas le seul paramètre sur lequel on puisse agir. La capacité des évaluateurs à trouver des erreurs, ou λ , peut aussi influencer favorablement le nombre total d'erreurs découvertes. Ce λ est fonction des compétences des évaluateurs. On distingue trois sortes d'évaluateurs :

- les novices, qui ne connaissent ni la tâche, ni l'ergonomie,
- les experts simples, qui connaissent l'ergonomie, mais pas la tâche,
- les doubles-experts, qui sont au courant à la fois de la tâche et de l'ergonomie.

Les novices trouvent en moyenne 22 % des erreurs chacun, contre 41 % pour les simples experts et 60 % pour les doubles-experts. Les doubles experts sont donc particulièrement recommandés quand il s'agit d'accomplir des performances optimales.

1.2. Les guidelines

Les guidelines sont des techniques qui sont un peu à mi-chemin entre les évaluations heuristiques et les inspections de standards. Il s'agit d'une inspection où des experts, ayant

établi une liste de guidelines à respecter, vérifient point par point l'adéquation entre l'interface et ces guidelines.

1.3. Les promenades pluridisciplinaires

Les promenades pluridisciplinaires réunissent trois types de personnes : des représentants des utilisateurs, des concepteurs et des experts en ergonomie. Des dessins, copies papier des écrans de l'interface sont ensuite présentés à chacun des membres du groupe. On demande ensuite à tous les participants d'effectuer une tâche prédéterminée, nécessitant de se déplacer dans l'interface et d'écrire toutes les actions à effectuer afin de mener la tâche à bien, et ce, de la façon la plus précise possible. Enfin, une discussion a lieu entre ces trois groupes sur base des réponses papier fournies.

1.4. Les inspections de cohérence

Les inspections de cohérence ont une approche inter-applicative puisque dans ce type d'inspection, des concepteurs de logiciels viennent vérifier la cohérence ergonomique de l'interface développée avec l'interface de leurs propres produits. De là, une liste de changements est rédigée afin d'assurer la cohérence de l'interface développée avec les autres. Le manque de cohérence d'une interface par rapport à elle-même peut être détectée à l'aide d'autres techniques, aussi bien informelles qu'empiriques (tests d'évaluation ou de validation, analyses heuristiques, inspections ergonomiques formelles,...).

1.5. Les inspections de standards

Les inspections de standards ont une préoccupation assez semblable. Il s'agit de mesurer la cohérence d'une interface par rapport à des standards pré-définis, que devraient respecter d'autres produits présents sur le marché. La grosse différence entre cette approche et la précédente se situe dans l'équipe qui est chargée de vérifier la cohérence. Dans ce cas, on n'utilise plus un groupe de développeurs mais un ou plusieurs experts en ergonomie.

1.6. Les promenades cognitives

Ce paragraphe se base sur l'article de C. Wharton, J. Rieman, C. Lewis et P. Polson "The Cognitive Walkthrough Method" [WHART94] extrait de l'ouvrage "Usability Inspections Methods" [MACK94].

Les promenades cognitives ont pour but d'évaluer une interface de façon à mettre en évidence la facilité avec laquelle les utilisateurs vont apprendre à s'en servir. Or, la facilité d'apprentissage dépend en grande partie de la facilité avec laquelle les utilisateurs vont pouvoir se promener dans l'interface sans se perdre, en pouvant revenir en arrière,... C'est donc à la lumière de la facilité d'exploration que l'interface sera jugée facilement 'apprenable' ou pas. Le processus d'évaluation par promenade cognitive se déroule en plusieurs étapes :

- la définition des inputs de la promenade,
- le rassemblement des analystes,
- la promenade proprement dite,
- la capture des informations critiques,
- la révision de l'interface.

1.6.1. La définition des inputs de la promenade

Il s'agit de fournir aux futurs analystes un certain nombre de données. Il s'agit typiquement de :

- une description détaillée de l'interface, qui doit comprendre tous les objets interactifs de l'interface ainsi que la réaction du système lorsque l'utilisateur activera ces objets,
- une description des utilisateurs, en terme d'expérience informatique et de background,
- une description des tâches que les analystes auront à mener à bien, qui doivent être représentatives de celles que les utilisateurs devront effectuer avec le système,
- une description de la séquence d'actions nécessaires à l'accomplissement de la tâche.

1.6.2. Le rassemblement des analystes

Les analystes qui vont mener à bien la promenade cognitive sont typiquement des concepteurs qui viennent de différents projets.

1.6.3. Déroulement de la promenade

Les analystes, à chaque étape de la progression de la tâche, essaient de voir quelles sont les actions possibles que le système offre, quelle serait la réaction des utilisateurs à ce moment, ce qu'ils essaieraient de faire afin de mener leur tâche à bien, s'ils disposent du feedback nécessaire en cas d'erreur,... En résumé, ils essaient de se mettre à la place de l'utilisateur essayant d'accomplir sa tâche. Ils suivent pour ce faire un modèle cognitif de résolution de problème, que les utilisateurs emploient typiquement quand ils interagissent avec l'interface. Ce modèle décompose les actions des utilisateurs en 4 étapes :

- les utilisateurs se représentent grossièrement la tâche qu'ils doivent accomplir,
- ils explorent l'interface dans le but de trouver un moyen d'accomplir la tâche, et tentent de mettre en oeuvre ces moyens,
- ils observent la réaction de l'interface pour évaluer les résultats des actions qu'ils ont entreprises à l'étape précédente,
- ils déterminent la prochaine action à accomplir

1.6.4. La capture des informations critiques

Il est intéressant de capturer les informations relatives à la promenade pour aider à la résolution des erreurs ergonomiques rencontrées. Pour ce faire, on peut effectuer des enregistrements vidéo, électroniques ou encore grâce à une simple prise de note manuelle.

Le processus d'inspection proprement dit est composé de 6 étapes :

- le planning, où le modérateur et le propriétaire s'assurent que les "inspecteurs vont bien disposer de toutes les informations nécessaires pour mener à bien leur tâche d'inspection" [MACK94, p.163]. Ces informations portent sur la conception du produit, les caractéristiques des utilisateurs et le scénario envisagé, ainsi que sur les heuristiques à appliquer dans la recherche des erreurs ergonomiques.
- la première réunion, où le modérateur et le propriétaire communiquent aux inspecteurs les informations issues de la première étape.

1.7.2. Le processus d'inspection

Les inspections ergonomiques formelles regroupent entre 4 et 8 participants, venant de différents horizons, avec divers backgrounds. Au sein de cette équipe d'inspection, les rôles sont bien définis. Sont désignés :

- un modérateur, qui dirigera les inspections et les réunions,
- un propriétaire, qui aura en charge l'implémentation des modifications de l'interface,
- des inspecteurs, qui essaieront de découvrir des problèmes ergonomiques,
- un secrétaire, qui notera tous les problèmes trouvés ainsi que les solutions apportées.

1.7.1. L'équipe d'inspection

1.7. Les inspections ergonomiques formelles

Nous nous baserons pour ce paragraphe sur l'article de M. Kahn et A. Prail "Formal Usability Inspections" [KAHN94] extrait de l'ouvrage "Usability Inspection Methods" [MACK94].

Il s'agit de modifier l'interface pour en éliminer les erreurs trouvées. C'est une étape tout-à-fait classique en évaluation de logiciels, et qui est à la base de toutes les méthodologies de conception itérative.

1.6.5. La révision de l'interface

- la préparation, où les inspecteurs essaient d'accomplir la tâche en jouant le rôle d'un utilisateur et en notant tous les problèmes ergonomiques rencontrés lors de l'exécution de la tâche demandée, ainsi que l'endroit où ces problèmes se trouvent.
- la seconde réunion voit la mise en commun de tous les problèmes détectés par les inspecteurs.
- le retravail où des solutions sont apportées aux problèmes et sont implémentées par le propriétaire, qui met donc à jour le produit.
- le suivi, où le modérateur rassemble des informations sur l'ensemble de l'inspection en vue de les faire examiner par d'autres équipes.

Pour réaliser l'inspection du produit, les inspecteurs, n'étant pas des experts en ergonomie, ont besoin de savoir que chercher et où le chercher. Il existe des méthodes pour les aider dans cette tâche.

a.) Que chercher

Savoir que chercher revient à comparer l'interface à la liste de règles ergonomiques fournie avant le test par le modérateur et le propriétaire. Elle s'inspire généralement des guidelines et heuristiques qui abondent dans la littérature.

b.) Où chercher

Savoir où précisément chercher ces erreurs requiert des inspecteur l'utilisation d'un modèle de performance de la tâche. Le schéma ci-dessous décrit ce modèle.

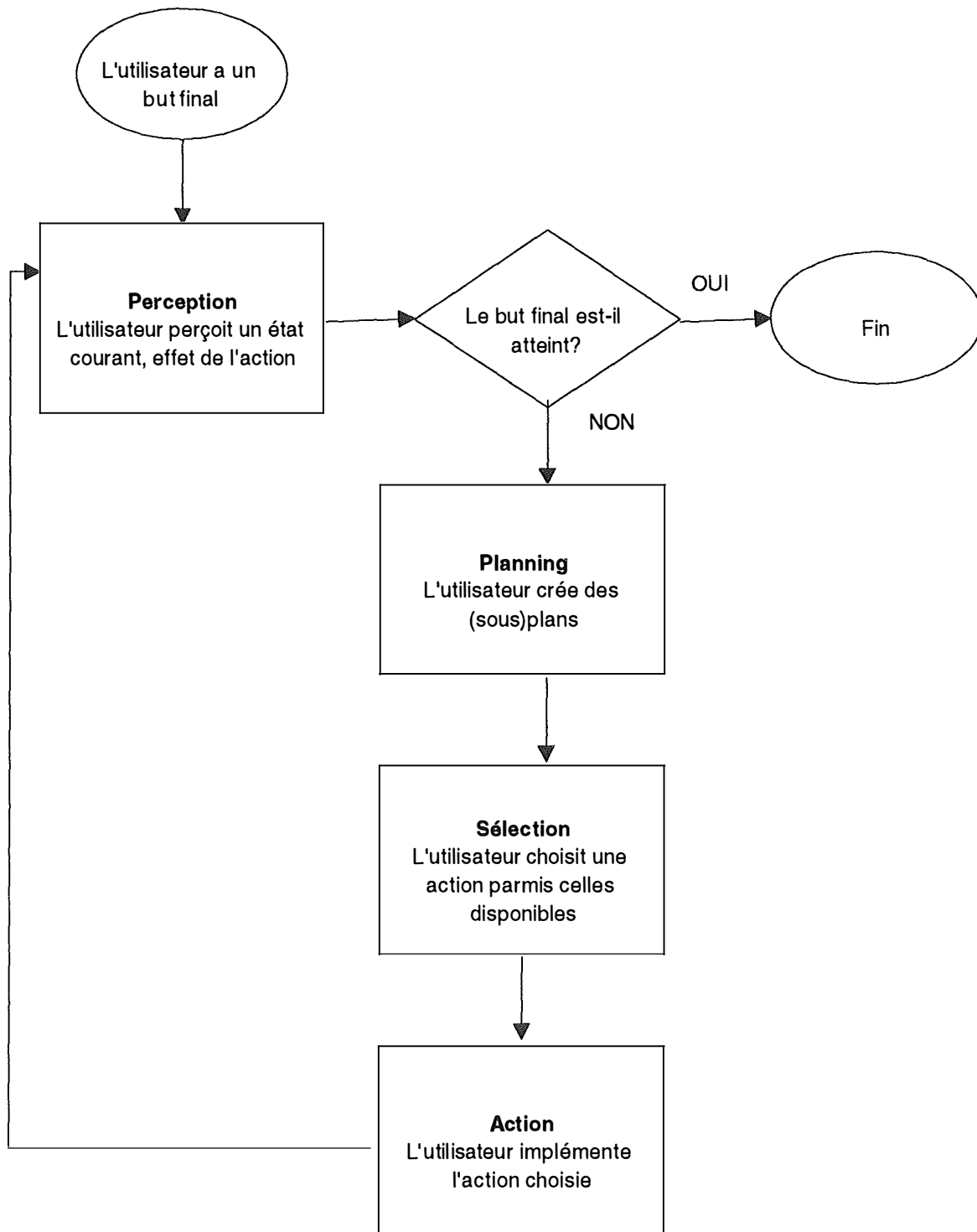


schéma 6 : le modèle de performance de la tâche [KAHN94, p.145]

A chaque étape sont associées des questions auxquelles doivent essayer de répondre les inspecteurs lorsqu'ils tentent de mener la tâche à bien. Au niveau de la perception, ils doivent se poser les questions suivantes :

- l'utilisateur verra-t-il clairement les informations requises?

- pourra-t-il facilement évaluer si le but final est atteint?
- dispose-t-il de suffisamment de feedback pour déceler ses erreurs?

Au niveau de la planification, le centre d'intérêt porte sur :

- quel modèle mental l'utilisateur est-il susceptible de mettre en oeuvre?
- comment l'utilisateur traduit-il le but?
- quelle est aide apportée à l'utilisateur pour l'aider à déterminer une planification correcte?

Au niveau de la sélection, le centre d'intérêt porte sur :

- l'utilisateur peut-il facilement identifier les options disponibles dans l'accomplissement de sa tâche?
- l'utilisateur peut-il facilement identifier les différences entre ces options?
- l'aide est-elle suffisante pour l'aider à ne pas se tromper dans la sélection d'actions?

Au niveau de l'action, le centre d'intérêt porte sur :

- quels sont les problèmes liés à l'accomplissement de la tâche?
- quels sont les actions du ressort du système et celles qui serait plus du ressort de l'utilisateur?
- combien de temps une action prend-elle?

1.8. Les inspections de caractéristiques

Les inspections de caractéristiques, quant à elles, se portent sur la fonction fournie par le système. On se pose ici la question du pourquoi des fonctions offertes, c'est-à-dire de l'utilité des fonctions fournies à l'utilisateur. Cette question est examinée sous deux angles : l'évaluation de cette fonction et sa conception.

2. Les évaluations formelles

Nous nous baserons pour ce paragraphe sur l'article de M. Gislin et C. Kolski [GRISL96]. Pour Gislin, les évaluations formelles sont des méthodes basées sur une modélisation formelle de ce qui devrait figurer dans l'interface, c'est-à-dire les besoins des utilisateurs en terme de "fonctionnalités, d'informations ou de formation complémentaire" [GRISL96, p.281]. Les méthodologies employées tendent en général à décomposer hiérarchiquement les tâches des utilisateurs en un ensemble de buts et sous-buts. Cette approche provient de deux sources :

- le génie logiciel,
- la linguistique.

Tous ces modèles peuvent être employés pour "prédire certains aspects liés à l'interaction homme-machine (hiérarchie des tâches, cheminement des actions de l'utilisateur, alternatives, temps requis pour réaliser une tâche...) et faciliter la mise en évidence des problèmes potentiels" [GRISL96, p.281]. Cependant, ces méthodes ne sont pas encore mûres pour réaliser une évaluation sur la qualité ergonomique même de l'interface. Les recherches dans cette voie se poursuivent.

Les modèles issus de la linguistique ont ceci de particulier, c'est qu'ils mettent en évidence la structure des interactions entre le système et les utilisateur grâce à une grammaire. Ils permettent donc de décrire les interactions interface-utilisateur, ce dont les modèles issus du génie logiciel étaient incapables, puisqu'ils se contentaient de décrire les tâches.

Toutes ces méthodes peuvent être utilisées pendant les tous premiers stades du cycle de développement du logiciel, c'est-à-dire dès les étapes d'analyse et de spécification.

3. Les évaluations automatiques

Contrairement aux approches formelles, où la modélisation de ce dont l'utilisateur avait besoin n'était pas informatisée, dans les évaluations automatiques, tout est effectué par un programme informatique. Le but de ces évaluations est d'ailleurs très différent. En effet, ils s'intéressent pour leur part à l'ergonomie des interfaces. Il est à remarquer que ces évaluations

sont encore du domaine de la recherche. On peut distinguer deux optiques dans l'évaluation automatique :

- les systèmes d'évaluation automatique de l'affichage
- les systèmes de génération automatique de l'affichage

3.4.1. Les systèmes d'évaluation automatique de l'affichage

Les systèmes d'évaluation automatique de l'affichage visent à mesurer l'ergonomie d'une interface sans aucune référence au contexte d'utilisation. Ils fourniront ainsi au concepteur la liste des erreurs ergonomiques flagrantes présentes dans les écrans, ainsi que la liste des améliorations éventuelles à amener. A noter que cette démarche ressemble fort à une automatisation pure et simple des méthodes de guidelines.

Parmi tous les systèmes d'évaluation automatique de l'affichage, on peut en citer deux qui semblent particulièrement prometteurs [GRISL96]. Il s'agit de :

- D.A.P. (Display Analysis Program),
- SYNOP

Ces systèmes illustrent deux philosophies de travail très différentes. Le système D.A.P. est basé sur un algorithme, capable de juger un écran sur base d'un ensemble de guidelines à respecter. Les préoccupations portent essentiellement sur le nombre d'informations présentées à l'utilisateur et devant être traitées par celui-ci, sur le nombre d'items par écran et sur la qualité de l'affichage.

Le système SYNOP, quant à lui, a recours à des techniques issues de l'intelligence artificielle pour juger de l'ergonomie des écrans. Il est également centré sur la présentation des informations dans l'interface.

Ces deux systèmes, utilisant des techniques assez éloignées pour juger de l'ergonomie d'une interface, aboutissent logiquement à des output sensiblement différents. Le système D.A.P. va en effet se contenter de générer une liste de recommandations à apporter aux écrans. Le système SYNOP va plus loin en corrigeant lui-même certaines des erreurs ergonomiques

trouvées. Cependant, dans le cas où cette correction s'avère difficile, il va fournir au concepteur des conseils sur la façon dont des changements pourraient être apportés.

Ces systèmes travailleront bien sûr vers la fin de la conception, une fois l'interface réalisée, en 'corrigeant' les écrans selon toute une série de critères et de règles ergonomiques.

3.4.2. Les systèmes de génération automatique de l'affichage

Il s'agit de "systèmes à base de connaissances ergonomiques visant la génération automatique d'affichage ou de spécification d'IHM" [GRISL96, p.286]. Une fois les écrans dessinés avec l'aide de ces systèmes, il faut encore consulter les utilisateurs sur ces écrans afin qu'ils puissent émettre leurs avis sur ceux-ci. Le but de ces systèmes est de produire des écrans (ou leurs spécifications) qui respectent les critères ergonomiques essentiels.

Comme nous l'avons déjà mentionné, ces outils sont encore à l'état de prototype de laboratoire. Cependant, on peut déjà estimer que certains de ces systèmes seront sans nul doute activement employés au cours des prochaines années. Les outils suivants, en particulier, semblent promis à un bel avenir [GRISL96] :

- ERGO-CONCEPTOR,
- A.P.T. (A Presentation Tool)

Ces deux outils poursuivent des objectifs différents. ERGO-CONCEPTOR, en effet, se base sur des techniques d'intelligence artificielle pour fournir au concepteur la liste des spécifications d'une interface sur base de la description interactive du produit en y incluant tout ce dont les opérateurs auront besoin. Le concepteur pourra, à l'aide de ces spécifications, réaliser les écrans de l'interface.

A.P.T., quant à lui, adopte la démarche inverse, c'est-à-dire que c'est sur la présentation des différentes informations devant figurer dans l'interface que se centre ce logiciel. Cet outil teste les différentes possibilités de présentation selon les guidelines classiques. Evidemment, il est toujours utile d'inclure suite à la génération d'un écran à l'aide de ce type d'outil un test avec des utilisateurs, pour être sûr que la présentation des informations réalisée par A.P.T. est bien conforme à ce qu'ils attendaient.

Ces systèmes, contrairement aux systèmes d'évaluation automatique, interviendront très tôt dans le cycle de développement d'un produit, c'est-à-dire dès le début de la phase de conception. Une fois les décisions prises de ce qui doit figurer dans les écrans, ceux-ci seront alors générés automatiquement.

Chapitre 6 :

Comparaison des méthodes d'évaluation de logiciels

Nous avons vu qu'il existe de nombreuses méthodes permettant d'effectuer une évaluation de l'interface. Il convient alors de se poser la question de savoir si toutes ces méthodes sont équivalentes, et si non, laquelle utiliser et dans quels cas.

A l'heure actuelle, tout le monde s'accorde à reconnaître les vertus des interactions utilisateurs-système lors de l'évaluation. La situation idéale reviendrait à faire effectuer des tests par des utilisateurs sur l'ensemble de l'application [NIELS93]. Mais ce n'est guère réaliste, dans la mesure où l'investissement à réaliser pour concevoir l'entièreté de l'interface serait trop important pour un simple test, qui, par ailleurs, devrait déboucher sur des modifications d'écrans,...Il faut donc souvent se contenter de faire tester un produit ou un prototype par un échantillon à choisir avec soin parmi la population d'utilisateurs.

Les méthodes automatiques, à l'heure actuelle, sont encore du domaine de la recherche. La démarche adoptée par les méthodes formelles est intéressante, car elle pourra fournir une aide précieuse aux développeurs. Cependant, pour le moment, elles se révèlent "extrêmement difficiles à mettre en oeuvre, tout en ne s'appliquant pas à des interfaces hautement interactives" [MACK94, p.2]. En conséquence, eu égard au fait que les approches formelles et automatiques sont encore du domaine de la recherche pour ce qui est de l'évaluation d'interface à proprement parler, nous ne les aborderons plus dans ce travail.

Quant aux méthodes informelles, il s'agit là de l'alternative la plus intéressante. En effet, toutes ne nécessitent pas la présence des utilisateurs et, en général, permettent de découvrir bon nombre d'erreurs ergonomiques. Est-ce à dire que ces méthodes sont appelées à remplacer progressivement les méthodes empiriques? Nous tâcherons de répondre à cette question au cours de ce chapitre.

D'un autre côté, nous avons vu que ces méthodes informelles étaient, pour la plupart, concurrentes, puisqu'elles ne sont pas clairement associées à un stade dans le cycle de développement d'un produit informatique. Avant donc de se lancer dans une comparaison entre les méthodes informelles et empiriques, il convient d'abord de comparer toutes ces

méthodes informelles entre elles, pour ne comparer que les plus efficaces avec les méthodes empiriques.

1. Comparaison des méthodes informelles

Les méthodes informelles sont nombreuses, et certaines peuvent avoir lieu au même endroit dans le cycle de développement du logiciel. Il faut donc déterminer laquelle de ces méthodes est la plus avantageuse, en quelles circonstances, et le pourquoi de cet état de fait.

Dans ce chapitre, nous n'évoquerons pas les inspections de caractéristiques, car elles se focalisent moins sur l'ergonomie d'une interface que sur l'utilité de telle ou telle fonction présente dans le système. Une telle approche, quoiqu'intéressante, n'est pas du ressort de ce travail. De même, nous n'aborderons pas en détail les guidelines, celles-ci n'étant que peu utilisées et constituant d'avantage un croisement entre deux techniques existantes qu'une technique à part entière.

Nous allons plutôt effectuer une comparaison des méthodes d'inspection susceptibles de se faire concurrence quand il s'agit de procéder à l'évaluation d'une interface, c'est-à-dire, d'une part, les promenades pluridisciplinaires, les promenades cognitives, les évaluations heuristiques et les inspections ergonomiques formelles, et d'autre part les inspections de standards et les inspections de cohérence . Mais pour mener à bien cette évaluation, il nous faut d'abord définir un ensemble de critères pour les juger.

1.1 Les critères d'évaluation de Karat

D'après Karat [MACK94], il existe des pistes (ou critères) pour comparer les différentes méthodes d'inspection. Ces pistes se basent sur la prise en compte des caractéristiques de l'inspection pour en évaluer l'efficacité dans un certain contexte. Il s'agit typiquement de :

- la manière de travailler des évaluateurs,
- leurs compétences,
- la façon dont ils explorent l'interface

1.1.1. La manière de travailler des évaluateurs

Les évaluateurs composant l'équipe d'inspection travaillent-ils individuellement ou en groupe? Karat recommande le travail en équipe pour tout ce qui touche à l'évaluation de tâches complexes. En effet, travailler en équipe permet de disposer d'un "large éventail de compétences et de backgrounds" [MACK94, p.222] nécessaire quand le problème ergonomique à résoudre est compliqué. Quand ce n'est pas le cas, travailler seul est préférable car la théorie psychologique a montré qu'il y a une tendance au nivellement vers le bas dans les groupes, ceux-ci travaillant "rarement au niveau des meilleurs de leurs membres" [MACK94, p.222].

1.1.2. Les compétences des évaluateurs

Traditionnellement, dans la littérature, on distingue plusieurs types d'évaluateurs. Les performances de l'équipe d'inspection seront d'ailleurs très variables selon ces différents types. Les évaluateurs peuvent en effet être des spécialistes en ergonomie, des développeurs ou encore des utilisateurs finaux du produit.

On appelle traditionnellement spécialiste un évaluateur qui a une connaissance approfondie de l'ergonomie cognitive et des guidelines présentes dans la littérature. De tels spécialistes étant difficiles à trouver, la tendance actuelle est d'enseigner les bases de ces disciplines à diverses personnes et d'ensuite considérer celles-ci comme des spécialistes relatifs. Dans la suite de cet exposé, nous appellerons par extension les personnes ayant une connaissance relative de l'ergonomie des 'spécialistes'.

Quand on parle de développeurs, on pense en premier lieu à ceux qui viennent de l'équipe de développement du produit à inspecter. En fait, il est vivement déconseillé de faire inspecter l'interface par les développeurs du logiciel. En effet, ceux-ci manquent du recul nécessaire et leur jugement risque de s'en trouver biaisé.

Quant aux utilisateurs, il s'agit classiquement de la population cible visée par le logiciel.

Maintenant, il est évident que si l'on a le désir de faire davantage participer des utilisateurs et des concepteurs à l'équipe d'évaluation (ce qui semble une suite logique au participatory design), mieux vaut utiliser des équipes pluridisciplinaires. Il s'agit là d'un choix politique, qui risque d'influencer les performances de l'évaluation. En effet, dans la littérature, on souligne les multiples avantages possibles si on implique des membres de l'équipe de développement dans les inspections. Par contre, des équipes de ce type sont beaucoup plus lentes que des équipes composées de doubles-experts, qui sont très efficaces, mais à qui on reproche parfois de ne pas s'intéresser assez aux utilisateurs et à leurs tâches.

1.1.3. La façon d'explorer l'interface

Les évaluateurs explorent-ils l'interface librement ou au contraire sont-ils soumis à la réalisation d'une tâche? Il apparaît que le choix entre ces deux options ne doit pas être guidé par des attentes de performance. En effet, celles-ci sont sensiblement équivalentes. L'utilisation de ces méthodes dépend en fait du contexte d'évaluation. Si "les tâches typiques d'un nouveau système sont connues et utilisées comme scénarii de base pour conduire le développement" [MACK94, p.226], alors il est préférable d'utiliser une inspection par promenade dans les tâches. Si on souhaite plutôt s'assurer de la cohérence et de l'ergonomie de la totalité de l'interface, mieux vaut employer l'exploration libre.

1.2. Application des critères aux promenades pluridisciplinaires, cognitives, aux évaluations heuristiques et aux inspections ergonomiques formelles

Toutes ces techniques sont concurrentes, et en tant que telles, doivent être comparées les unes aux autres. En fait, quand on les examine de près, on peut voir qu'elles sont très différentes les unes par rapport aux autres suivant les critères de Karat.

Les promenades pluridisciplinaires comprennent typiquement un groupe d'évaluateurs provenant de divers domaines et explorant l'interface de manière non autonome.

Les promenades cognitives, pour leur part, comprennent un groupe de concepteurs qui explorent l'interface de manière un peu particulière. Ils se promènent dans celle-ci à la manière d'un utilisateur, c'est-à-dire qu'étant donné une action à accomplir, ils essaient de prendre des chemins identiques à ceux que prendrait l'utilisateur. Il n'est donc pas exclu qu'ils se perdent dans l'interface s'ils pensent que l'utilisateur ferait de même.

Les évaluations heuristiques reposent idéalement sur l'avis d'un ou plusieurs experts libres de parcourir l'interface comme bon leur semble. Ces experts doivent travailler de façon indépendantes.

Les inspections ergonomiques formelles, pour finir, sont constituées de personnes venant de différents horizons, explorant l'interface de façon à exécuter une tâche donnée.

Si l'on compare le nombre d'erreurs trouvées selon les différentes méthodes, on s'aperçoit que les performances de celles-ci peuvent être très différentes. Un exemple de cet état de fait est donné par le tableau suivant, extrait de l'ouvrage "Usability Inspection Methods" [MACK94, p.184]. Il s'agit d'une comparaison entre les résultats obtenus par la méthode d'évaluation heuristique et ceux obtenus par la promenade cognitive, lors de l'exécution de six tâches à l'aide d'une interface basée sur l'utilisation du téléphone. Six groupes composés de trois personnes ont évalué l'interface selon les deux méthodes. Les personnes impliquées se décomposaient en outre selon trois degré d'expertise :

- les utilisateurs novices, qui ne connaissent ni la tâche, ni l'ergonomie,
- les développeurs, qui connaissent la tâche, mais pas l'ergonomie,
- les experts, qui connaissent l'ergonomie, mais pas la tâche.

Les résultats sont ensuite comparés à ceux trouvés en effectuant des évaluations empiriques auprès de 18 utilisateurs, et ceci afin de déterminer le pourcentage d'erreurs découvert par les méthodes informelles par rapport à l'ensemble des erreurs trouvées par testage.

Les résultats de cette comparaison sont à prendre avec prudence, vu le peu de mesures effectuées (un seul groupe pour chaque cellule du tableau). Ils sont donc donné à titre purement illustratif, puisque nous ne pouvons tirer de conclusions fiables d'un aussi faible

échantillon. Cependant, ces résultats ont le mérite de mettre en lumière une tendance qui se trouve confirmée par les critères de Karat.

Méthode	Evaluateurs	Pourcentage de problèmes trouvés	Améliorations proposées
Evaluations heuristiques	Experts en ergonomie	44 %	77 %
	Développeurs	16 %	3 %
	Utilisateurs	8 %	6 %
Promenades cognitives	Experts en ergonomie	28 %	16 %
	Développeurs	16 %	3 %
	Utilisateurs	8 %	6 %

schéma 7 : nombre de problèmes ergonomiques trouvés et améliorations proposées suite à une enquête menée auprès de six groupes d'évaluateurs [DESURV94, p.184]

L'évaluation heuristique, d'après cette expérience, semble donc donner de meilleurs résultats que la promenade cognitive, du moins dans la détection de problème. En ce qui concerne la sévérité des erreurs, des résultats similaires furent notés [MACK94, p.185]. Comme nous l'avons déjà dit, les différences observées sont sujettes à controverse, le nombre de mesures prises étant insuffisant. Néanmoins, ces résultats, aussi partiels soient-ils, correspondent à l'application des critères de Karat relativement à ces deux méthodes d'évaluation.

Méthode	Evaluateurs	Problèmes cosmétiques	Problèmes provoquant des erreurs	Problèmes provoquant des erreurs
Evaluations heuristiques	Experts en ergonomie	80 %	67 %	29 %
	Développeurs	40 %	0 %	12 %
	Utilisateurs	20 %	0 %	6 %
Promenades cognitives	Experts en ergonomie	40 %	67 %	18 %
	Développeurs	0 %	0 %	12 %
	Utilisateurs	20 %	0 %	6 %

*Schéma 8 : Les erreurs ergonomiques découvertes classées en fonction de leur gravité
[DESURV94, p.185]*

En ce qui concerne les autres méthodes d'inspection, les promenades pluridisciplinaires et les inspections ergonomiques formelles, nous ne disposons pas de données chiffrées quant à leurs performances respectives. C'est pourquoi nous allons utiliser les critères de Karat.

Nous avons vu que les promenades cognitives fournissaient des résultats assez décevants par rapport aux évaluations heuristiques. Une explication de cet état de fait peut venir des critères de Karat. En effet, les évaluations heuristiques diffèrent tout d'abord des promenades cognitives au niveau de la liberté d'exploration. Car on sait que s'il s'agit d'évaluer la totalité d'une interface, l'exploration libre est recommandée. Or, une des ambitions affichées des promenades est de vérifier la facilité pour les utilisateurs de se retrouver dans l'ensemble de l'interface. De plus, dans les promenades cognitives, les évaluateurs sont des concepteurs. Il ne s'agit donc pas de spécialistes en ergonomie, comme dans les évaluations heuristiques. Pour terminer, les analystes travaillent en groupe pour les promenades cognitives contre un travail individuel avec mise en commun pour les évaluations heuristiques. Mais dans ce dernier cas, on pourrait se poser la question de la complexité de l'évaluation. Il semble que l'évaluation d'une interface d'une certaine complexité serait alors plus du ressort des promenades cognitives. Celles-ci sont d'ailleurs nées de la demande d'avoir des méthodes qui puissent évaluer la facilité de se retrouver dans des interfaces complexes (c'est typiquement le cas de l'hypermédia). Donc, malgré les résultats empiriques et théoriques (deux critères favorables contre un seul) qui semblaient trancher favorablement en faveur de l'évaluation heuristique, il

ne faut pas perdre de vue que les promenades ont un rôle à jouer, surtout dans les cas complexes.

Les promenades pluridisciplinaires, quant à elles, diffèrent des promenades cognitives du fait que les évaluateurs composant le groupe d'inspection proviennent de divers domaines. Il semblerait que ce facteur soit encore plus favorable à l'évaluation de situations complexes puisque cela élargit l'éventail de compétences et de backgrounds. D'ailleurs, "de meilleurs résultats sont obtenus quand un des analystes a un background en psychologie cognitive" [MACK94, p.140]. En conséquence, les promenades pluridisciplinaires me paraissent devoir être plus efficace que les promenades cognitives, en se situant sur le terrain de la résolution de problèmes complexes. Cependant, comme les promenades pluridisciplinaires se basent sur des dessins d'écrans, on peut se poser la question de leur efficacité plus tard dans le cycle de développement, une fois que des prototypes sont disponibles. En dehors de ce cas, il me paraît plus avantageux d'utiliser les promenades pluridisciplinaires. Par rapport aux évaluations heuristiques, qui rencontrent deux critères à la place d'un pour les promenades, ces dernières restent préférables pour les évaluations complexes où leur efficacité a déjà été mise en lumière.

Les inspections ergonomiques formelles quant à elles, utilisent des personnes d'horizons différents, travaillant individuellement et mettant en commun leurs résultats dans une optique d'accomplissement de la tâche. Pour les évaluations d'interface peu compliquées, cette méthode a les avantages des évaluations heuristiques, puisque les évaluateurs travaillent indépendamment. Le contexte des inspections est la découverte d'erreurs ergonomiques dans l'accomplissement des tâches prescrites, ce qui diffère du point de vue des évaluations heuristiques. Une autre différence avec les évaluations heuristiques se situe dans l'expertise des évaluateurs. On sait que ces évaluateurs viennent de différents domaines, ce qui peut constituer un avantage, vu les backgrounds différents, dans les situations plus complexes. Mais pour les situations simples, les experts (et surtout les doubles experts) sont incontestablement plus rapides. En conséquence, les inspections ergonomiques formelles sont des méthodes à utiliser lorsque les interfaces sont de complexité moyenne, mais que l'on désire néanmoins être tournés vers les utilisateurs et leurs tâches.

Graphiquement, on peut résumer les décisions relatives à l'utilisation de telle ou telle méthode de la façon suivante :

Préoccupations \ Critère	Manière de travailler	Compétence de l'équipe	Manière d'explorer
Interface très complexe	en groupe	pluri-disciplinaire avec ergonomes	selon la tâche
Evaluation de la totalité de l'interface	seul	ergonomes	librement
Conception participative	en groupe	pluri-disciplinaire avec ergonomes et utilisateurs	selon la tâche

Schéma 9 : tableau des critères de Karat et des préoccupations lors de l'évaluation

1.3. Application des critères aux inspections de standards

et inspections de cohérence

Tout d'abord, il est à remarquer que les inspections de standards et de cohérence sont des techniques qui s'inscrivent dans une optique un peu spéciale puisqu'il s'agit de mesurer la cohérence de plusieurs interfaces. Ces méthodes ne se focalisent pas sur l'ergonomie d'un produit, et donc ne peuvent être utilisées afin de deviner quels seront les problèmes d'utilisation rencontrés par les utilisateurs.

Nous pouvons déjà remarquer qu'une différence entre les inspections de standards et de cohérence réside dans la composition du groupe d'inspection. On peut dès lors se poser la question de savoir quel groupe est le plus efficace à mener à une modification des incohérences, de celui qui est composé de développeurs (inspection de cohérence) ou de celui qui est composé d'un ou plusieurs experts (inspection de standards).

D'après D.Wixon et al. [Mack94], il semblerait que ce soit le premier groupe qui soit le plus à même d'établir la liste des changements à apporter. Les experts "pressentent que les check-lists sont utiles pour les problèmes simples, mais que les problèmes plus complexes nécessitent un dialogue avec des concepteurs expérimentés" [MACK94, p. 88]. D'autre part, il

Un troisième point qui pourrait, lui, jouer en faveur des inspections de standards est la constatation que si on souhaite examiner la cohérence et l'ergonomie de la totalité de l'interface, mieux vaut employer l'exploration libre. Or, dans ce cas bien précis, il faut examiner l'ergonomie de toute l'interface pour pouvoir la comparer aux autres. Il semblerait que cette

Une seconde différence entre les deux méthodes réside dans la façon de travailler. Les inspections de cohérences utilisent des groupes de concepteurs travaillant ensemble, les inspections de standards un ou plusieurs experts en ergonomie qui mettent leurs résultats en commun au cours d'une réunion. Or, pour faire face à des problèmes complexes (et la génération de recommandations de changements en est un), mieux vaut faire appel à un groupe.

Dans les inspections de cohérences, les évaluateurs connaissent les tâches de l'utilisateur et donc disposent d'un champs de discussion plus vaste.

Dans les inspections de standards, on ne se préoccupe pas du tout des tâches de l'utilisateur, les experts ne les connaissant pas et dès lors se contentant de scanner l'interface.

Ces résultats sont à mettre en rapport avec les critères vus plus haut. A choisir entre une équipe d'experts et une équipe pluridisciplinaire, mieux vaut opter pour la seconde si l'on se trouve dans une situation de conception de l'interface où l'on désire se centrer d'avantage sur les utilisateurs et leurs tâches.

A la fin des inspections de cohérence donnaient de bons résultats. A la fin des réunions, les changements à apporter étaient clairement indiqués, tandis que les incohérences "pour lesquelles il n'y a pas de remède simple sont notées pour un travail de conception futur" [MAC94, p. 90]. Il est donc préférable d'utiliser les inspections de cohérences car en faisant intervenir les développeurs, des décisions concernant les changements à apporter sont prises en connaissance de cause, c'est-à-dire que les concepteurs voient mieux ce qu'il est possible de changer dans l'immédiat, à l'opposé des experts qui ne peuvent que constater telle ou telle incohérence sans avoir la moindre idée de la difficulté de mise en oeuvre des modifications nécessaires. De plus, ils font des recommandations pour le développement de produits futurs, ce dont les experts étaient incapables.

démarche soit plus efficace que celle des inspections de cohérences qui progressent pas à pas dans l'interface.

Néanmoins, d'après les critères de Karat (deux critères favorables pour les inspections de cohérence contre un seul pour les inspections de standards) et les observations sur le terrain [Mack94, chapitre 4], les inspections de cohérence semblent nettement plus efficaces que les inspections de standards.

2. Comparaison des méthodes d'évaluation empiriques et des méthodes d'évaluation informelles

Nous avons examiné l'efficacité dont les différentes méthodes d'évaluation informelles concurrentes pouvaient faire preuve. Il reste à voir si ces méthodes peuvent concurrencer et même supplanter les méthodes très populaires pour le moment issues de l'évaluation empirique.

Tous les auteurs, cependant, aussi bien les tenants de l'évaluation empirique que les adeptes de l'évaluation informelle sont d'accord pour dire que non, ces deux méthodes s'avérant plus complémentaires que concurrentes, agissant dans des circonstances d'évaluation assez différentes.

Reste alors à se poser la question de savoir quand et pourquoi utiliser l'évaluation informelle en lieu et place de l'évaluation empirique. Nous nous baserons sur l'ouvrage de Jeffrey Rubin "Handbook of usability testing : how to plan, design and conduct effective tests" [RUBIN94] et de Robert L.Mack et Jakob Nielsen "Usability Inspection Methods" [MACK94].

2.1. Les limites de l'évaluation empirique

L'évaluation empirique passe dans la littérature pour être en général la meilleure approche des problèmes ergonomiques. "Quand elle est conduite avec soin et précision, c'est un indicateur quasi infallible des problèmes potentiels et des moyens pour les résoudre"

[RUBIN94, p.27]. D'ailleurs, les tenants des méthodes informelles jugent de l'efficacité de leurs méthodes par rapport aux résultats obtenus grâce à une évaluation empirique.

Cependant, les évaluations empiriques ne sont pas toujours la meilleure technique à utiliser. Tout d'abord, elles sont très gourmandes en termes de temps passé à tester l'interface. Une expérience a été menée pour démontrer la lenteur des évaluations empiriques [MACK94, p.212]. Deux groupes furent formés afin de tester une interface. L'un d'entre eux était composé de deux ergonomes conduisant un groupe de six utilisateurs. L'autre regroupait seulement quatre ergonomes. Le premier groupe effectua l'évaluation empirique du produit en 33.2 heures, le second en à peine quatre heures. Le temps d'évaluation est donc un point faible des évaluations empiriques.

De plus, elle est chère en utilisateurs. En effet, l'idée de faire effectuer régulièrement des tests par des utilisateurs entre des modifications mineures de l'interface revient à gaspiller cette ressource utilisateur pourtant précieuse puisque le testeur idéal ne devrait jamais avoir été confronté au produit. Comme le souligne Rubin, dans les premières étapes de l'élaboration de l'interface, il se peut qu'il y ait des violations flagrantes des règles ergonomiques élémentaires. Il n'est dans ce cas pas utile de faire effectuer des tests par des utilisateurs à ces stades de la conception, pour "découvrir ce qui est évident" [RUBIN94, p.27].

Bien sûr, cette constatation semble remettre en cause l'utilité et l'efficacité des tests exploratoires, et surtout des tests d'évaluation. Les tests exploratoires ont pour but de mieux comprendre les processus de pensée des utilisateurs. Il ne s'agit donc pas de mener une évaluation d'interface mais plutôt de comprendre comment en écrire une qui s'adapte au mieux avec la façon de penser des futurs utilisateurs du système. Il n'y a donc pas de concurrence entre cette approche et les évaluations informelles, qui elles ont pour objet l'évaluation ergonomique de l'interface. Par contre, les tests d'évaluation sont, eux, nettement concurrents. En effet, ils portent sur l'évaluation ergonomique d'un prototype assez détaillé. Il faut donc se poser la question de savoir s'il n'est pas préférable d'utiliser les évaluations informelles à ce stade de développement du produit. Pour Nielsen, mais aussi pour Rubin, il semblerait qu'effectivement, les évaluations informelles soient moins coûteuses, moins gaspilleuses d'utilisateurs et soient suffisamment efficaces que pour remplacer les tests d'évaluations.

2.2. Les limites de l'évaluation informelle

Les évaluations empiriques ont la réputation de donner des résultats extrêmement complets et fiables pour découvrir des erreurs ergonomiques. Mais d'autre part, les évaluations informelles sont moins coûteuses à la fois en temps et en utilisateurs. Nous avons vu les cas où il était avantageux de recourir à des évaluations informelles, eu égard aux limitations des évaluations empiriques. Mais les méthodes informelles ne peuvent pas toujours être utilisées en lieu et place des évaluations empiriques, ayant elles-mêmes certaines limitations. Nous allons décrire ces limitations au cours de ce paragraphe suivant.

2.2.1. Définition des objectifs d'évaluation

Un point faible des évaluations informelles réside dans la difficulté à assigner des objectifs d'évaluation. Or, trouver un maximum d'erreurs ergonomiques n'est pas toujours le seul objectif à prendre en considération. Il peut y en avoir d'autres, par exemple s'assurer que "80 pourcents des utilisateurs seront capables de mener à bien les tâches de l'application sans avoir besoin d'assistance", ou encore que "les utilisateurs seront capables de mener à bien la tâche X en moins de 10 minutes" [MACK94, p.205]. Il s'avère que les méthodes d'inspection ne sont pas capables de prendre en compte de tels objectifs. En conséquence, lorsque l'on doit décider quelle méthode d'évaluation utiliser, il faut d'abord se poser la question des objectifs que l'on désire assigner à cette évaluation. S'il ne s'agit que de vérifier l'ergonomie d'une interface, alors les méthodes informelles peuvent constituer une alternative intéressante aux méthodes empiriques. Si l'on souhaite s'assurer des taux d'accomplissement de la tâche sans erreur, du temps passé à mener une tâche à bien,...alors mieux vaut utiliser les méthodes empiriques.

2.2.2. Nombre et type d'erreurs trouvées

La plupart des auteurs sont d'accord pour dire que le nombre d'erreurs et le nombre d'erreurs sérieuses découvertes est nettement plus important quand on effectue des tests avec les futurs utilisateurs.

2.2.3. Génération de recommandations

Les évaluations empiriques sont reconnues pour leur lenteur quand il s'agit de tester une interface. Cependant, il pourrait y avoir des nuances à apporter en termes de temps passé à interpréter les résultats de ces tests pour élaborer des stratégies de changement de l'interface. En effet, si on prend une méthode d'évaluation informelle, par exemple l'évaluation heuristique ou la promenade cognitive, on s'aperçoit que ces méthodes donnent des résultats satisfaisant pour ce qui est de détecter des erreurs ergonomiques. Quant il s'agit de les corriger, elles sont nettement moins efficaces du fait que les inspecteurs n'ont pas de contact avec les utilisateurs. Ces derniers ne peuvent donc leur communiquer d'informations sur le contexte d'utilisation de l'application. De plus, le nombre de problèmes trouvés est nettement plus élevé lors d'une évaluation empirique que lors d'une évaluation informelle. Or, note Karat, "au plus on détecte d'erreurs pour une portion de l'interface qui pose problème, au plus la source de données pour générer des recommandations sur cette portion de l'interface est riche" [MACK94, p.218]. Il semblerait donc que la source de renseignement dont on pourrait tirer des recommandations est nettement plus faible dans le cas de l'évaluation informelle.

2.2.4. Evaluation du produit avant sa diffusion

Les évaluations informelles disposent en leur sein de méthodes capable de scanner efficacement une interface dans son entièreté (par exemple, les évaluations heuristiques). Cependant, il semble souhaitable de ne pas diffuser un produit avant d'avoir pu effectuer une évaluation empirique de celui-ci. En effet, aussi bien les tenants des évaluations informelles que des évaluations empiriques déconseillent l'usage des évaluations heuristiques, promenades cognitives,...tout à la fin du cycle de développement du produit, c'est-à-dire juste avant sa diffusion. Bien sûr, si une évaluation empirique n'était pas possible à ce moment précis (coût, disponibilité d'utilisateurs,...), alors mieux vaudrait effectuer une évaluation informelle que pas d'évaluation du tout. Cependant, si une évaluation empirique est possible, alors il est vivement conseillé d'en réaliser une. Tout d'abord, le nombre d'erreurs ergonomiques trouvé est nettement plus important. Ensuite, parce que la génération de recommandations et l'évaluation des objectifs doivent être facilités à ce stade du développement.

Tous ces arguments font pencher la balance en faveur d'une évaluation empirique avant la diffusion du produit. Les tests de validation, tests empiriques qui ont lieu tardivement dans le cycle de développement du produit semblent parfaitement adaptés pour remplir ce rôle. Quant aux évaluations informelles, comme nous l'avons vu, elles ne peuvent se justifier à cette étape que s'il n'était pas possible d'effectuer des tests de validation.

3. Intégration des différentes méthodes d'évaluation au cours du cycle de développement

Il y a de multiples avantages à effectuer des évaluations tout au long du cycle de développement d'un logiciel. Ces premières évaluations donnent un feedback sur la qualité de l'interface utilisateur à un moment où effectuer de gros changements de celle-ci est toujours possible et pas trop coûteux.

Une démarche possible serait d'appliquer les évaluations de la façon suivante dans le cycle de développement :

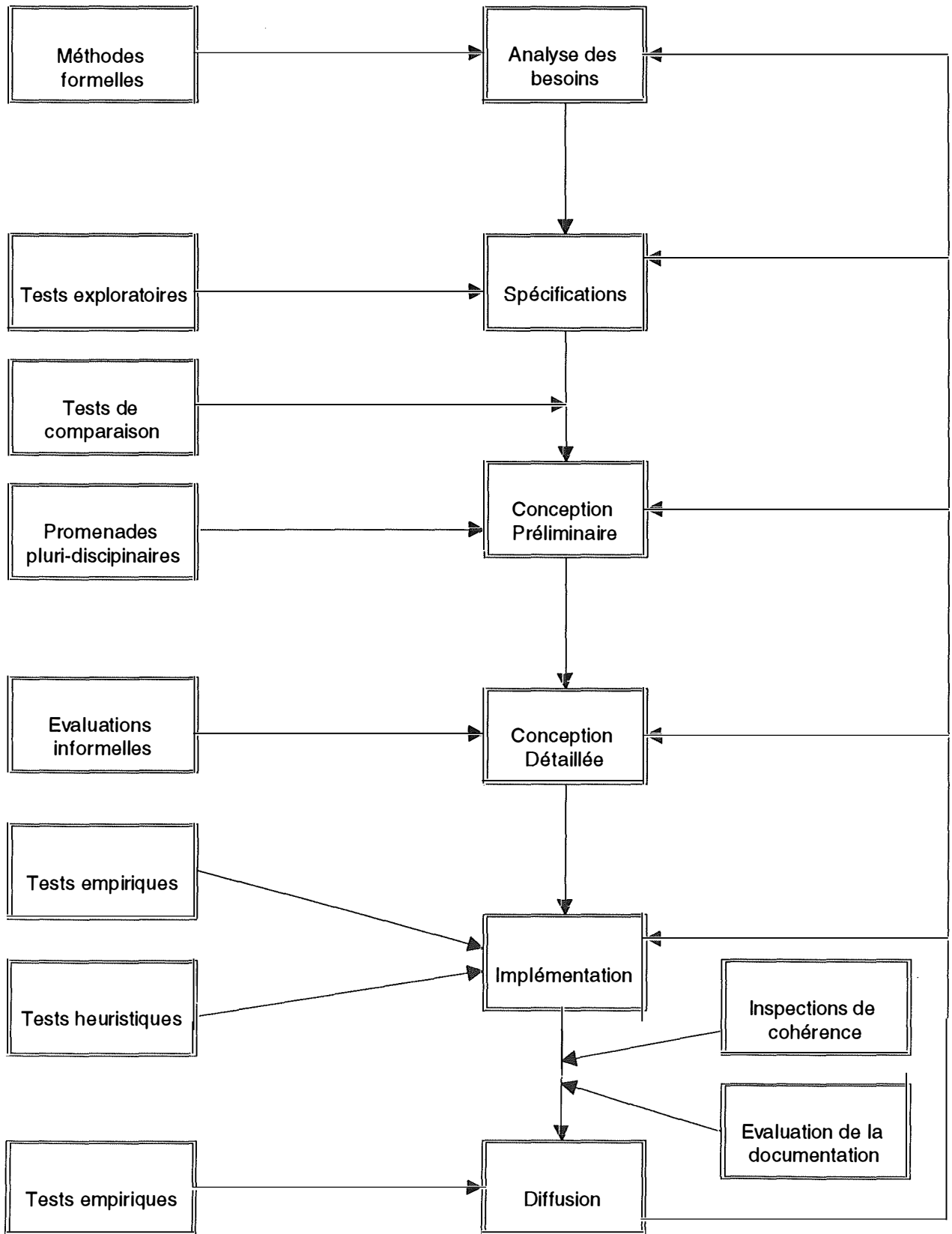


Schéma 10 : Proposition d'intégration des méthodes d'évaluation le long du cycle de développement du logiciel

Avant même la rédaction des spécifications, il peut être utile de mieux comprendre la façon de penser des utilisateurs. Ainsi que le préconisent des auteurs tels Nielsen et Rubin, une idée intéressante serait de faire participer un ou plusieurs utilisateurs à l'équipe de projet. D'un autre côté, les méthodes formelles peuvent servir ici, puisqu'il ne s'agit pas d'effectuer une évaluation ergonomique mais de mieux cerner les utilisateurs et tout ce dont ils ont besoin pour mener à bien les tâches.

Lorsqu'on se trouve à l'étape de rédaction des spécifications, il convient de s'assurer que les développeurs aient correctement interprété la façon de penser des utilisateurs. Pour vérifier que toutes les hypothèses faites au sujet de l'utilisateur sont correctes, il faudrait faire en sorte que les utilisateurs "réfléchissent tout haut" sur l'accomplissement de leurs tâches, et que ces réflexions soient à la disposition de l'équipe de projet. Les tests exploratoires, menés auprès de quelques utilisateurs, peuvent très bien remplir cette mission. Ce sont eux qui vont fournir à l'équipe de projet le feedback nécessaire à s'assurer de la bonne compréhension des processus mentaux des utilisateurs.

Au niveau des spécifications, Rubin, tout comme Nielsen, plaident en outre pour les tests de comparaison, qui portent sur différents styles d'interface, pour découvrir quel serait le plus approprié pour le produit développé. Autrement dit, on se servirait de l'évaluation de plusieurs interfaces pour recommander l'utilisation de tel ou tel objet interactif, de telle ou telle métaphore, qui sembleraient plus ergonomiques que d'autres. L'idée d'utiliser des tests à ce niveau paraît bonne, car les méthodes informelles sont reconnues pour ne pas être appropriées quand il s'agit de générer des recommandations.

Au stade de la conception préliminaire, on travaille sur des dessins d'écrans. Les promenades pluridisciplinaires sont bien adaptées à ce genre de situation, les autres évaluations fonctionnant en général mieux plus tard dans le cycle de développement, idéalement avec un prototype. Quant aux évaluations empiriques, elles sont trop gourmandes en utilisateurs que pour être employées ici, alors qu'il n'y a pas encore d'interface à proprement parler. Il faut remarquer que l'équipe de projet ne se coupe pas des utilisateurs en employant les promenades pluridisciplinaires. En effet, ces méthodes préconisent que des représentants des utilisateurs participent à la promenade.

Au niveau de la conception détaillée, Rubin préconisait les tests d'évaluation pour détecter les erreurs ergonomiques et bien comprendre comment l'utilisateur effectue sa tâche. Bien sûr, des prototypes devaient être disponibles. Des évaluations de ce type semblent cependant assez lourdes à ce niveau, et les évaluations informelles paraissent tout-à-fait capables de les remplacer. On peut ici utiliser, si l'interface n'est pas trop complexe, les inspections ergonomiques formelles qui permettent de rendre en compte des tâches et des problèmes ergonomiques. Si au contraire l'interface était très compliquée, avec des risques pour l'utilisateur futur de s'y perdre, on pourrait alors utiliser les promenades cognitives, en y incluant si possible un expert en ergonomie.

Lors de l'implémentation, il faut vérifier que l'interface est bien construite et qu'elle ne contient plus d'erreurs ergonomiques. A ce niveau, c'est-à-dire peu avant la diffusion du produit, il est fortement conseillé d'utiliser des méthodes d'évaluation empirique, en raison de la difficulté d'évaluation des méthodes informelles à ce moment du cycle de développement. Ces carences des méthodes informelles ont été vues au point 3.2.4. de ce chapitre. Une fois les erreurs ergonomiques trouvées, on peut modifier l'interface en fonction de celles-ci. A noter qu'entre les changements apportés à l'interface, il devient inutile de faire des tests empiriques, vu le peu de modifications apportées. Une évaluation heuristique suffit alors amplement.

Avant la diffusion, il est intéressant d'effectuer une inspection de cohérence, afin que le produit soit cohérent avec d'autres logiciels du marché. Les modifications qui résulteront de ces inspections pourront à nouveau être évaluées par des heuristiques.

Une étape supplémentaire à réaliser à ce niveau, et dont on parle habituellement assez peu, concerne le manuel et la documentation fournie avec le logiciel. Il s'agit pourtant d'un aspect important. En effet, il y a des logiciels qui sont par nature compliqués et qui nécessitent une certaine connaissance des utilisateurs. Le manuel est alors là pour aider l'utilisateur à faire ses premiers pas dans l'interface et à ne pas se perdre dans celle-ci. La documentation, définie comme l'ensemble des informations à donner sur le produit et ses conditions d'utilisation, peut donc s'avérer d'une aide précieuse [MEDA90].

Nous nous baserons principalement pour l'évaluation de la documentation d'un logiciel sur la grille d'analyse de la méthodologie MEDA [MEDA90]. Il s'agit dans cette optique de répondre à un certain nombre de questions ayant trait à tout ce qui touche à la documentation et au manuel d'utilisation du produit. Parmi ces questions, les plus intéressantes sont reprises ci-dessous :

- le manuel contient-il des explications précises sur l'utilisation du logiciel? L'utilisateur doit en effet savoir où commencer, quelles sont les options qui lui sont offertes et commencer se déplacer dans l'interface.
- le manuel contient-il des références à de la littérature supplémentaire à consulter?
- la documentation précise-t-elle bien les pré-requis nécessaires avant d'utiliser le produit?
- le manuel contient-il des exemples d'utilisation du logiciel?
- le manuel est-il facile à lire, clair et bien rédigé?
- le manuel est-il pratique, facile à utiliser?

Tous ces points clés dans l'évaluation de la documentation permettent de vérifier facilement la qualité d'un manuel avant sa mise sur le marché.

Une fois le produit diffusé, ou lors de l'évaluation d'un logiciel existant, il s'agit de récolter des données sur la façon dont il est utilisé, s'il est bien accepté des utilisateurs,... Les évaluations empiriques semblent appropriées à ce niveau.

Chapitre 7 :

Développement et évaluation de didacticiels

Les méthodes d'évaluation de l'ergonomie des logiciels sont nombreuses et s'appliquent à différents endroits du cycle de développement. La question de l'évaluation se pose pour l'E.A.O. dans une optique un peu différente. En effet, il ne suffit plus de déterminer la facilité d'utilisation du logiciel, il faut également que celui-ci soit porteur d'une valeur pédagogique.

L'Instructional System Development préconise comme première étape à la conception de matériel didactique la définition des objectifs. Ces objectifs portent sur les compétences que l'on voudrait voir atteindre par les étudiants. Le degré de réalisation de ces objectifs est mesuré par les performances des étudiants face au didacticiel, et en cas d'inadéquation entre les performances voulues et effectives, le didacticiel sera modifié.

Cette approche par raffinements successifs est commune à de nombreux modèles de conception, aussi bien de didacticiels que de logiciels. Cependant, tous les auteurs ne sont pas d'accord d'effectuer ces raffinements sur base des performances des utilisateurs. Tout d'abord, l'idée de faire tester par les étudiants le système réel est peu économique. Cela sous-entend que le didacticiel doit être terminé pour pouvoir commencer à effectuer des tests. Ensuite, la façon de mener les tests relèvent de l'évaluation empirique, avec la lenteur et le gaspillage d'utilisateurs que cela implique. Enfin, ils sont peu adaptés lorsqu'il n'y a que des modifications mineures à apporter à l'interface.

1. Intégration des méthodes d'évaluation de logiciels dans les étapes de conception d'un didacticiel classique

La plupart des méthodologies de conception et développement de didacticiels sont inspirées du modèle cybernétique et de l'I.S.D. Mais au fil des ans, les méthodologies se sont progressivement détachées du modèle de base de l'I.S.D. pour adopter une vision plus moderne d'évaluation continue tout le long du cycle de développement.

Un exemple de cette évolution peut être illustré en étudiant la méthodologie proposée par R. Besnainou, C. Muller et C. Thouin dans l'ouvrage "Concevoir et utiliser un didacticiel". Nous l'avons déjà abordé au chapitre 1 (cfr. chapitre 1, paragraphe 2.2.). Nous allons maintenant examiner plus en détail cette méthodologie, et essayer de voir quels sont parmi les différents types d'évaluation vus, ceux qui pourraient s'intégrer dans les étapes de développement du didacticiel. Pour ce faire, il convient tout d'abord d'étudier quelles sont les différentes étapes dans le développement d'un produit éducationnel.

1.1. La méthodologie classique

Le cycle de développement d'un didacticiel, d'après Besnainou, est illustré par le schéma ci-dessous :

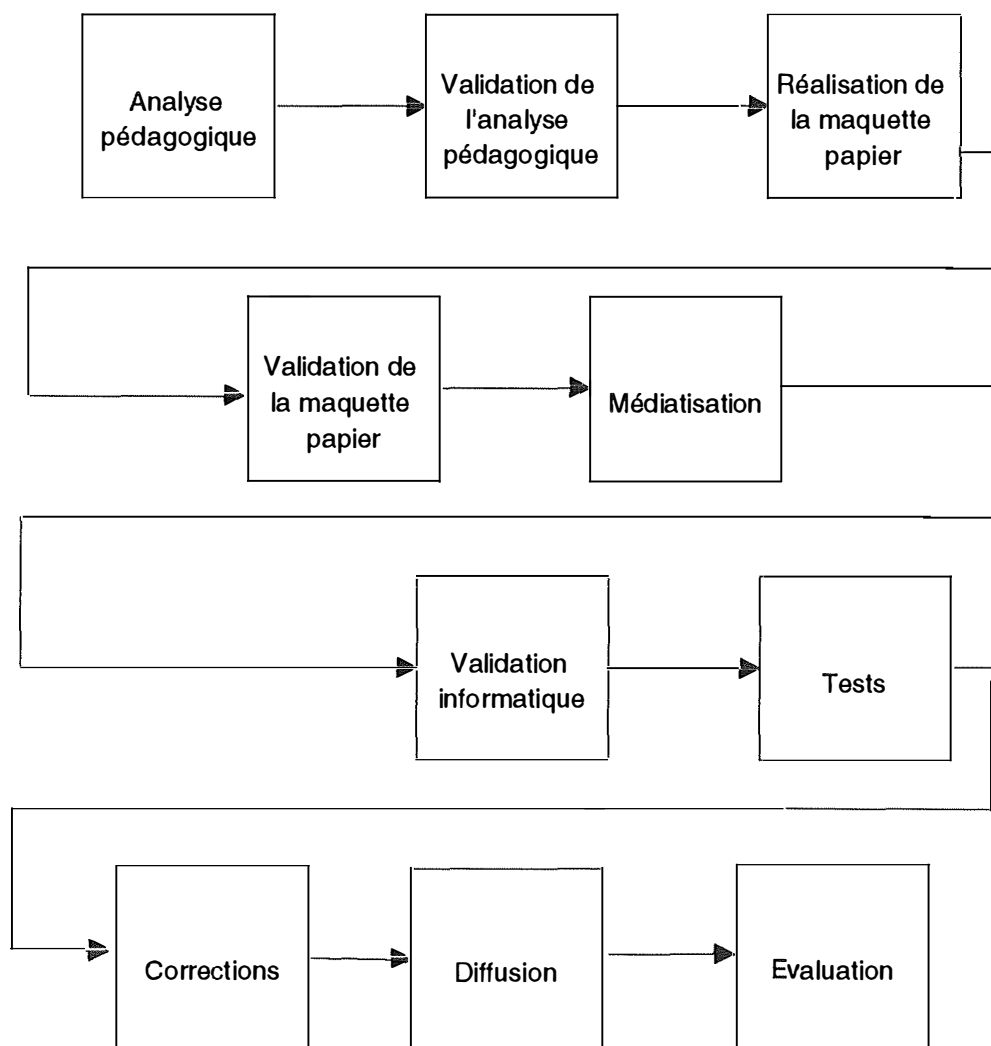


Schéma 11 : Les étapes de développement d'un didacticiel [BESN88, p.26]

1.1.1. Analyse pédagogique

L'analyse pédagogique est une étape essentielle, parce que c'est à ce niveau que des décisions clés seront prises quant à la suite du développement. Cette analyse repose elle-même sur 3 étapes :

- définition de l'objectif pédagogique,
- analyse de la population cible,
- définition de la stratégie pédagogique.

a.) La définition de l'objectif pédagogique

L'objectif pédagogique, en accord avec la vision I.S.D., est la première étape du cycle de développement du didacticiel. Il s'agit de décider des capacités que les étudiants devront acquérir grâce au didacticiel, ainsi que de fixer l'étalon de mesures qui permettra de juger si l'enseignement a été efficace.

b.) L'analyse de la population cible

Tout comme dans les méthodologies de développement de logiciel, une attention particulière est apportée à la caractérisation des futurs utilisateurs du système et de l'environnement dans lequel ils évolueront.

c.) La définition de la stratégie pédagogique

Mener une stratégie , c'est choisir comment déployer ses ressources en vue d'atteindre un objectif. En E.A.O., il faut faire toute une série de choix avant d'entamer l'étape suivante, la réalisation de la maquette papier. Ces choix portent sur 4 grands types de ressources :

- la ressource matériel, c'est-à-dire le hardware, le software, ainsi que les manuels et la documentation disponibles pour soutenir le didacticiel. Les choix porteront sur la façon de "gérer l'écran, utiliser les couleurs,..." [BESN88, p.37] Les décisions prises ici auront une incidence lors de la réalisation de la maquette papier.

- la ressource contenu, c'est-à-dire que les choix à faire porteront sur les matières à aborder dans le cours, celles à laisser tomber et la manière de structurer ces matières entre elles.
- la ressource élève, définie sur base de l'analyse de la population cible effectuée à l'étape 1.1.2. Les choix porteront tout d'abord sur les activités d'apprentissage à présenter aux étudiants afin qu'ils acquièrent les connaissances convenues lors de la définition de l'objectif pédagogique. Ensuite, il restera à définir le scénario et la mise en scène des activités.
- la ressource unité d'interaction consiste en la définition des unités d'interaction, qui sont la "structure de base de l'échange d'informations" entre l'étudiant et l'ordinateur [BESN88, p.22]. Les choix porteront sur la façon d'agencer ces unités et sur les formes d'interaction à attribuer pour telle ou telle activité, la forme d'interaction étant le style utilisé pour faire faire un exercice à l'élève et s'assurer ainsi qu'il a bien acquis un concept.

1.1.2. Validation de l'analyse pédagogique

Besnainou voit dans cette étape le feu vert du commanditaire sur le projet. Celui-ci peut ainsi faire les remarques qui lui semblent pertinentes sur l'analyse pédagogique et marquer son accord. L'étape est importante pour Besnainou : "On n'a pas intérêt à négliger cette étape sauf à courir le risque de voir le travail remis en cause quand toute la maquette papier a été réalisée" [BESN88, p.27].

1.1.3. Réalisation de la maquette papier

La réalisation de la maquette papier peut être vue comme la conception préliminaire dans le cycle de développement de Rubin. La maquette papier est rédigée sur base des décisions prises lors de l'analyse pédagogique. Il est évident qu'au cours de cette réalisation, es concepteurs doivent s'efforcer de suivre un certain nombre de recommandations ergonomiques citées dans la littérature, telles celles portant sur le placement des informations sur un écran, ou celles sur une utilisation judicieuse des couleurs,...

1.1.4. Validation de la maquette papier

Besnainou recommandait que la maquette papier soit testée auprès d'un échantillon d'utilisateurs futurs du système. Cependant, le centre d'intérêt de cette étape n'est pas la qualité de l'interaction ou l'ergonomie du produit. Comme le fait remarquer Besnainou, on s'intéresse ici "aux problèmes de fond" [BESN88, p.28].

1.1.5. Médiatisation

La médiatisation recouvre deux étapes qui se trouvent séparées dans la majorité des méthodologies de développement de logiciels ou de didacticiels, la conception détaillée et l'implémentation informatique.

1.1.6. Validation informatique

Besnainou voyait dans cette étape le testage du didacticiel de façon à ce que celui-ci ne contienne plus de bugs. Cette étape, quoique purement informatique, est bien sûr importante et doit être menée avant que les tests avec les élèves ne commencent. Ces derniers ne doivent en effet n'être utilisés que pour la découverte d'erreurs ergonomiques, pas pour découvrir des problèmes d'implémentation.

1.1.7. Tests

Ils sont menés auprès d'un échantillon représentatif des étudiants. Ils occupent une place prépondérante dans la vision I.S.D. de conception itérative sur base des performances.

1.1.8. Corrections

Des corrections sont apportées au didacticiel sur base des performances des étudiants et des erreurs ergonomiques résiduelles.

1.1.9. Diffusion

La diffusion a trait à la "stratégie globale d'insertion de l'E.A.O." dans l'organisation [BESN88, p.29].

1.1.10 Evaluation

L'évaluation, en tant qu'étape du processus de développement vise à mesurer l'adéquation entre les objectifs définis à l'étape 1 et les performances des étudiants. La différence entre cette étape et l'étape 1.7. réside dans le fait qu'ici, on s'intéresse aux performances de tous les utilisateurs en milieu de travail réel. Ainsi que le fait remarquer Nielsen, cette étape est très importante pour le "développement de nouvelles versions ou de nouveaux produits" [NIEL93, p.109].

1.2. Intégration de l'évaluation ergonomique et de concepts issus du génie logiciel

La méthode de Besnainou, Muller et Thouin peut être enrichie d'un certain nombre de concepts qui proviennent des méthodologies de développement de logiciels. Ces apports sont de deux types :

- méthodologiques, c'est-à-dire qu'ils concernent des ajouts ou des modifications dans la façon dont s'enchaînent les étapes de développement du didacticiel,
- évaluatifs, c'est-à-dire qu'ils concernent la façon dont les évaluations pourraient être menées et planifiées dans le didacticiel.

Le schéma ci-dessous montre les différentes étapes qui pourraient être utilisées dans le développement d'un didacticiel

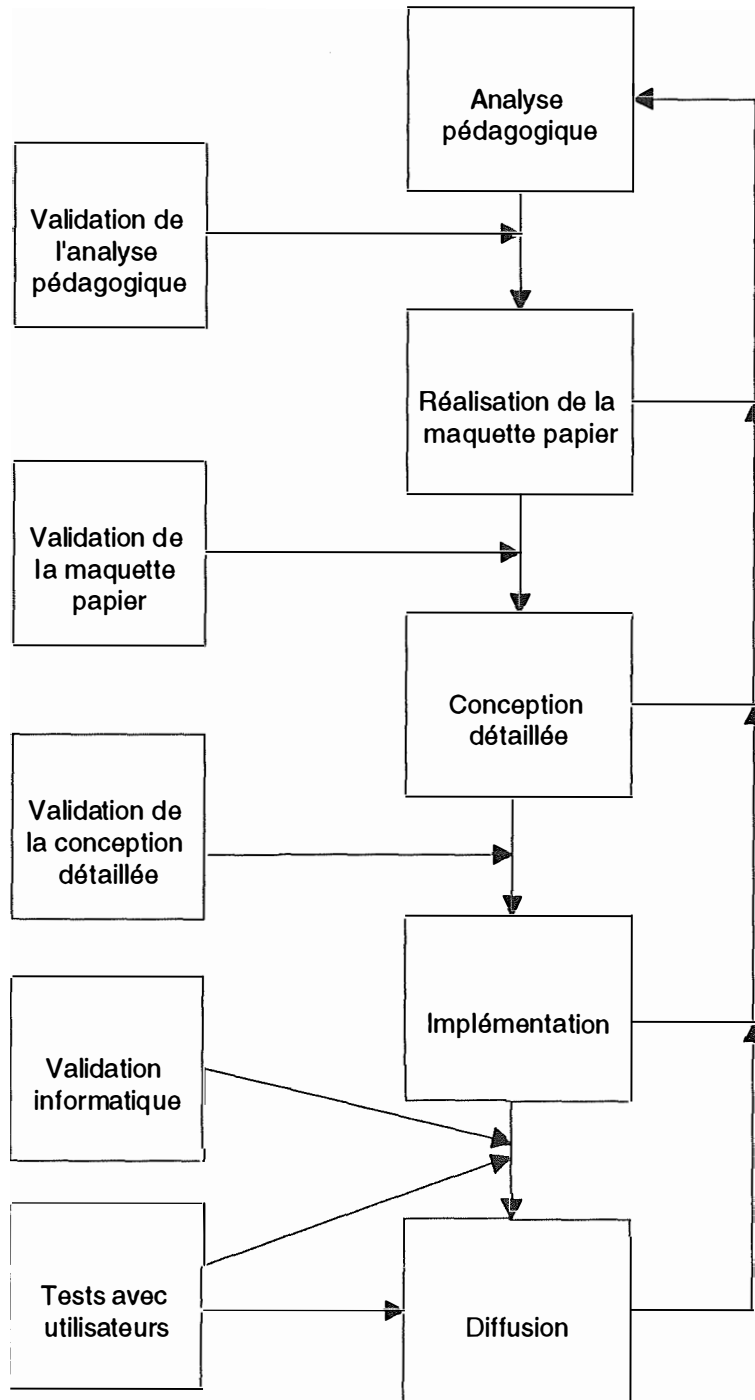


Schéma 12 : Intégration de l'optique d'évaluation continue dans le cycle de développement d'un didacticiel

1.2.1. Les apports méthodologiques

On constate que plusieurs étapes ont été ajoutées par rapport au schéma original. Besnainou n'envisageait qu'une seule étape, la médiatisation, pour traiter de la conception détaillée et de l'implémentation. Cette idée ne me semble pas heureuse. Avec cette manière de procéder, des tests ne peuvent plus avoir lieu entre les étapes de conception détaillée et d'implémentation, comme c'était le cas dans d'autres méthodologies. L'évaluation sur prototype est oubliée. Or l'évaluation sur prototype est une évaluation qui peut se faire grâce à des méthodes informelles, comme les évaluations ergonomiques formelles ou les promenades cognitives. Ces évaluations étaient relativement peu coûteuses et permettaient de trouver un certain nombre d'erreurs ergonomiques. Si ces erreurs ne sont pas découvertes avant l'implémentation du produit, le seul moment où on pourra les découvrir sera lors des tests finaux avec les utilisateurs. C'est très tard, et demandera aux concepteurs de modifier plus profondément l'interface que si on avait déjà pu détecter et remédier à un certain nombre de problèmes.

Sans être remise en question, l'analyse pédagogique, et en particulier la caractérisation de la population cible pourrait cependant être enrichie de concepts présents dans des méthodologies de développement de logiciels. L'idée, que nous avons déjà abordé au cours du chapitre 3, de faire participer un étudiant à l'équipe de projet semble intéressante, et pourrait être le signe d'une évolution importante dans la manière de créer du matériel instructionnel.

1.2.2. Les apports évaluatifs

Lors de l'étape de validation de l'analyse pédagogique, Besnainou insistait sur la nécessité d'obtenir une certaine confirmation que ce qui était décrit jusque là correspondait bien à la réalité et à ce que l'organisation désirait en terme de didacticiel. Mais pour obtenir cet aval, il s'adresse au commanditaire du projet. Il est vrai que d'un point de vue organisationnel, on ne puisse se passer de son appui. Mais limiter l'aval au seul commanditaire me paraît risqué. Mieux vaudrait en effet avant de se lancer dans l'écriture de la maquette papier trouver confirmé les hypothèses faites quant aux utilisateurs et à leur façon de penser. Des tests exploratoires menés auprès de quelques étudiants pourraient éclairer sous un jour nouveau les connaissances et processus de pensée de ces derniers.

L'idée de réaliser la validation de la maquette papier n'est en soi pas mauvaise, mais dans l'optique de Besnainou, elle est uniquement vue dans une optique "problèmes de fonds". C'est-à-dire que les concepteurs réalisent le test dans une optique très pédagogique. Pourquoi ne pas profiter de ces tests pour se pencher également sur des problèmes de formes? De cette façon, on pourrait avoir une évaluation ergonomique plus continue à travers les différentes étapes de réalisation du didacticiel. D'un autre côté, le terme 'tests avec les utilisateurs' fait fortement penser à de l'évaluation empirique, avec les défauts sous-jacents à cette méthode (temps de mise en oeuvre long, gaspillage d'utilisateurs). Une évaluation informelle, telle la promenade pluridisciplinaire semble une alternative intéressante à appliquer dans ce cas, d'autant plus que cette évaluation travaille justement avec des dessins d'écrans.

Par contre, l'idée d'effectuer une évaluation empirique après la validation informatique est très intéressante. Elle correspond à l'idée de nombreux auteurs de faire effectuer des tests après l'étape d'implémentation. Mais ainsi que le souligne Nielsen, seules les modifications importantes méritent d'être re-testées auprès des utilisateurs. Lorsqu'il s'agira de changements mineurs à apporter, les méthodes d'évaluations informelles suffiront amplement.

En résumé, on peut adapter le schéma d'évaluation continue des logiciels présenté au chapitre 5 de la façon suivante :

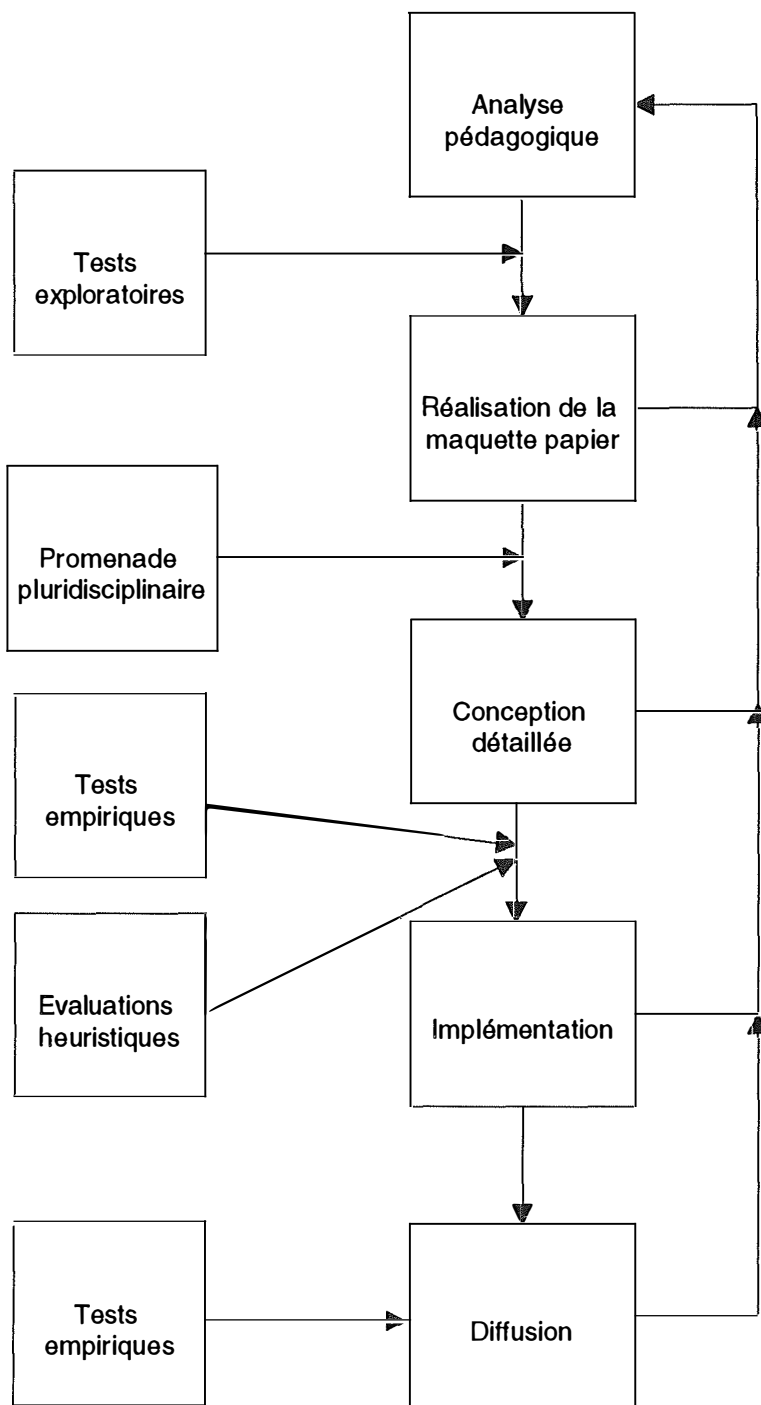


Schéma 13 : Intégration des méthodes d'évaluation dans le cycle de développement d'un didacticiel

2. Le cas particulier des didacticiels hypermédia

Il se dégage parmi les auteurs une grosse majorité pour dire que le développement des didacticiels hypermédiés doit suivre une philosophie I.S.D. Cependant, la spécificité du médium n'est pas sans répercussions sur la façon de mettre au point et d'évaluer les didacticiels. En fait, des méthodologies rigoureuses prescrivant un modèle de développement de didacticiels hypermédiés n'existent pas encore, étant toujours du domaine de la recherche.

Cependant, des auteurs ont identifié les facteurs à prendre en compte lors du développement de tels systèmes. C'est notamment le cas d'A. Romiszowski, qui explicite ces facteurs dans son article "The Hypertext/Hypermedia Solution - But What Exactly is the Problem?" [ROMIS90].

2.1. Ajouts lors de la mise au points de didacticiels

A. Romiszowski décompose les méthodologies I.S.D. en quatre niveaux de prises de décisions :

- le supra système, ou système client, constitue le premier niveau de décision,
- le système instructionnel, c'est-à-dire un cours, un programme d'enseignement, se situe au deuxième niveau
- les composants du système instructionnel, c'est-à-dire tout ce qui touche aux plans de leçons, d'exercices,...représente le troisième niveau
- les matériaux utilisés pour soutenir le cours et les plans de leçons sont au quatrième niveau de prise de décision.

Les ajouts relatifs à l'utilisation de l'hypermédia comme support vont se diviser et se répartir selon ces quatre niveaux.

2.1.1. Le supra système

Le supra système est composé de l'ensemble des pressions sociales, organisationnelles, politiques, philosophiques,... qui vont faire pencher la balance dans le sens de certaines solutions par rapport à d'autres.

Les systèmes hypertextes ne font pas exception à la règle. Au contraire, il existe des courants d'idées, fondées ou non, qui poussent à l'adoption de tels systèmes. A. Romiszowski cite, par exemple l'idée que "si la connaissance est un pouvoir, alors au plus nous avons de connaissances à notre portée (par un click souris), au plus notre société sera puissante et prospère, de même que les individus qui y vivent" [ROMIS90, p.330]. De même, des pressions économiques, politiques peuvent s'exercer pour pousser à l'adoption de systèmes hypermédia.

Ce type de considérations supra système ne sera pas abordé davantage, car il relève plus de la sociologie, de l'étude des organisations,... que de l'informatique, même si son influence sur le choix du médium est déterminant.

2.1.2. Le système instructionnel

Le deuxième niveau de l'I.S.D. donnera en output un cours complet. Le système instructionnel est caractérisé par la prise de décision quant à la conception du système au sens large. Les décisions à prendre concernant le système sont de décisions de haut niveau, générales, qui concernent la totalité du cours. On appelle parfois ce second niveau le niveau stratégique. Ce niveau correspond en gros à l'analyse pédagogique décrite par Besnainou [BESN88] et à la conception préliminaire décrite par Rubin [RUBIN94].

L'hypermédia impose à ce niveau toute une série de décisions supplémentaires à prendre. A. Romiszowski cite dans son article les douze points suivants [ROMIS90, p.334] :

1.) But : quel sera le but du système ? Servira-t-il comme simple support à la recherche d'informations, ou les étudiants pourront-ils l'annoter, créer leurs propres modules et liens? c'est également à ce niveau que l'on fixe, comme en I.S.D. classique, l'objectif du système.

- 2.) Tâches/contenu : le concepteur va-t-il se limiter à relier différents documents qui existaient déjà , va-t-il aller plus loin en les modifiant ou en les annotant, ou au contraire devra-t-il générer tout le système à partir de rien?
- 3.) Caractéristiques de l'utilisateur : le système va-t-il s'adresser à une seule catégorie de couples utilisateurs/objectifs, pourra-t-on définir plusieurs catégories de couples utilisateurs/objectifs, ou le système pourra-t-il être accessible à tous les utilisateurs?
- 4.) Mode d'utilisation : le mode d'utilisation dérive immédiatement du point 1. Le système ne permettra-t-il que de lire ou l'ajout de commentaires et la création de nouveaux modules seront-ils admis?
- 5.) Besoins par rapport au support : les modules seront-ils hypermédia, purement textuels ou combineront-ils texte, image et son?
- 6.) Taille des modules : des décisions doivent être prises sur la taille des modules contenant de l'instruction. Divisera-t-on la matière en chapitres, en pages?
- 7.) Liens du réseau : ils vont déterminer toute la structure du réseau. S'agira-t-il d'une structuration orientée modules (on relie les modules entiers entre eux) ou orientée concepts (on relie les concepts des modules pour former un réseau sémantique)?
- 8.) Représentation du réseau : utilisera-t-on un modèle statique ou dynamique?
- 9.) Liberté de navigation : l'étudiant sera-t-il libre de browser où il le désire, lui fournira-t-on une aide sous forme de conseil à la navigation, ou au contraire sera-t-il purement et simplement dirigé par où le concepteur désire qu'il aille?
- 10.) Aide à la navigation : quelle sorte d'aide sera fournie à l'utilisateur, une aide en ligne, un index ou une aide "intelligente"?
- 11.) Environnement de conception : de quel type d'outils informatiques, de quel environnement les concepteurs disposeront-ils?
- 12.) Approche d'évaluation : sur quelle base reposera l'évaluation, sur la satisfaction subjective de l'utilisateur, ou sur des critères spécifiques tels les performances?

Ces différentes décisions s'articulent selon le schéma suivant :

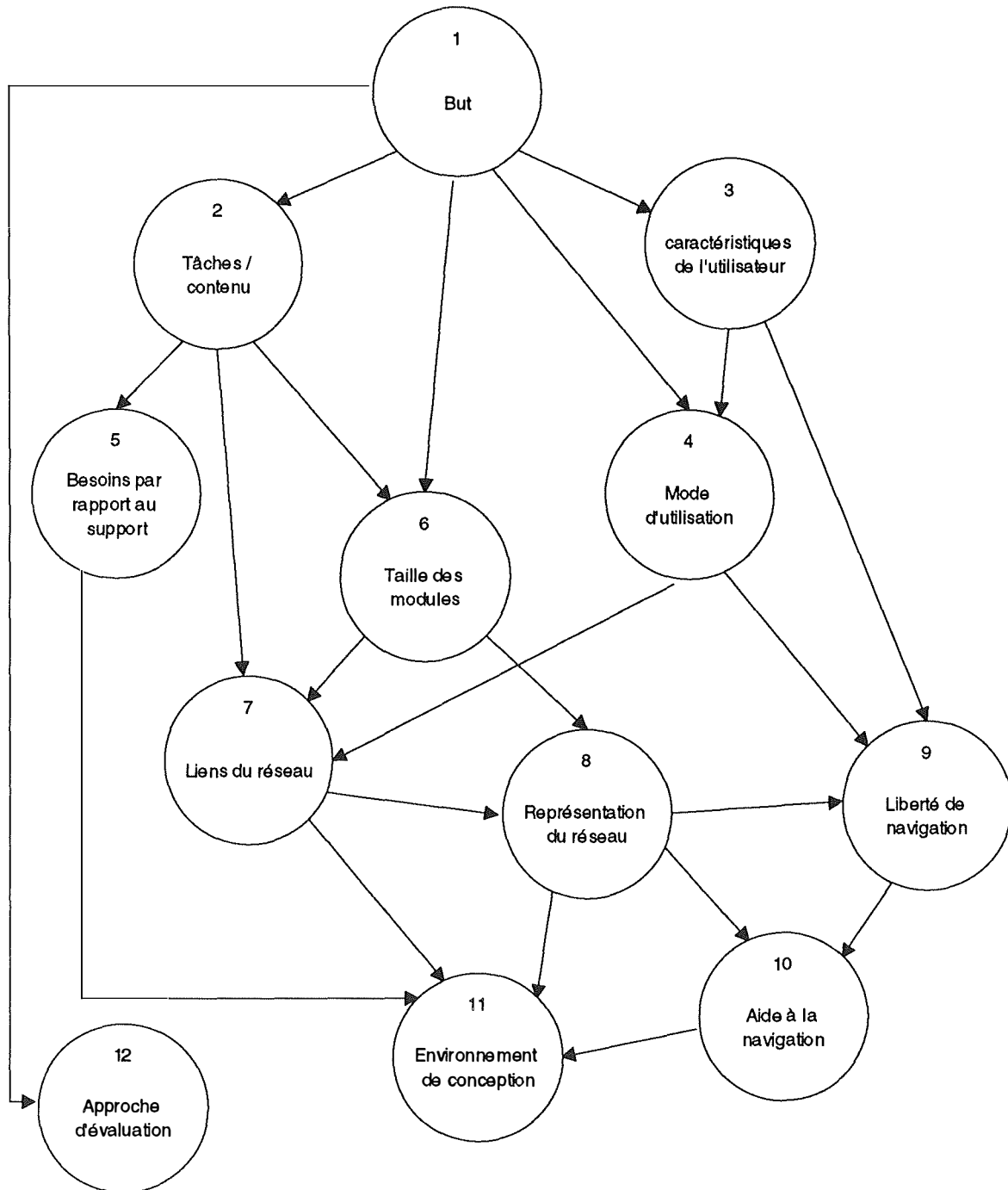


schéma 14 : Réseau des considérations de conception [ROMIS90, p.339]

Il faut remarquer que ce schéma sous forme de réseau est dynamique, les nœuds représentant les questions de développement de systèmes hypermédia. Les liens représentent quant à eux les principales interactions entre les différents éléments. Le numéro des nœuds indique "l'ordre dans lequel les décisions devaient idéalement être prises" [ROMIS90, p.338].

Les évaluations pour didacticiels hypermédia proposées dans la littérature sont quasiment toutes du ressort de ce second niveau. Elles s'attardent sur l'estimation du rapport inputs/outputs, en accord avec la philosophie I.S.D. Les détails du système hypermédia, quant à eux, sont vus à travers une approche black-box.

2.1.3. Les composants du système instructionnel

Ce niveau correspond en gros à la conception détaillée et à l'implémentation de Rubin. C'est à ce niveau que sont prises en I.S.D. les décisions concernant les leçons constituant le cours et les spécifications du matériel didactique destiné à l'enseignement de ces leçons. On appelle parfois ce troisième niveau le niveau tactique.

En ce qui concerne les didacticiels hypermédiés, c'est à ce niveau que les concepteurs mettent en place le réseau et ses composants. Les scénarii sont planifiés, les modules sont écrits, de même que les liens entre ces modules.

Tous ces éléments font partie de la stratégie globale, découlant des décisions prises au second niveau. Les aides à la navigation seront pour leur part écrites à la fin de cette conception détaillée, c'est-à-dire une fois que la structure du réseau sera connue.

Il est à remarquer qu'il n'existe pas encore de méthodes universellement reconnues pour décrire et enchaîner les étapes du niveau tactique, certains auteurs insistant en effet sur l'importance de l'intuition et de l'expérience passée dans ce domaine. Cependant, A. Romiszowski, tout en soulignant les divergences existant dans la littérature, propose une description et un enchaînement des étapes du niveau 3. Il en distingue 4 :

- analyse des objectifs correspondants à chaque leçon et dont l'ensemble doit correspondre à l'objectif global du système.
- applications des principes et guidelines à la conception d'une maquette du système.
- développement de cette maquette, aussi poussé que possible mais sans avoir recours à des techniques spécialisées de production, comme l'intégration à la maquette de séquences vidéo,...

- évaluation de la maquette. Cette évaluation, d'après A. Romiszowski, peut être réalisée par des tests empiriques, les avis d'autres développeurs,...Il semble judicieux de faire cette évaluation au moyen des promenades cognitives, particulièrement recommandées dans les cas complexes où il faut mesurer la facilité de l'utilisateur à se retrouver dans une interface.

Ces étapes semblent assez classiques en I.S.D. et pourraient être intégrées à la méthodologie de développement de didacticiels décrite au point 1.2. de ce chapitre.

2.1.4. Les matériaux utilisés

Ce dernier niveau de décision en I.S.D. est caractérisé par "le travail très détaillé et spécialisé nécessaire pour livrer aux étudiants le produit final" [ROMIS90, p.348]. Ce niveau correspond donc aux tests informatiques et aux ultimes évaluations empiriques du produit.

L'hypermédia apporte à ce stade du développement des préoccupations spécifiques dont les concepteurs se doivent de tenir compte. A. Romiszowski en distingue 4 :

- la cohérence du texte compris dans les modules : il faut vérifier que le texte d'un module puisse être lu indépendamment des modules vus précédemment.
- l'opportunité d'utiliser des graphiques pour réaliser des "communications de grandes quantités d'informations interreliées" [ROMIS90, p.348], étant donné que la taille maximale des modules imposée par certains systèmes.
- la sélection des média est importante, puisqu'il devient possible de "passer d'un médium à un autre en l'espace de quelques secondes" [ROMIS90, p.348]
- l'utilisation efficace des outils d'aide au développement et à la création de systèmes hypermédia.

2.2. Evaluation des didacticiels hypermédia

L'évaluation des didacticiels hypermédia a été étudiée entre autre par J. Nielsen [NIELS90] dans son article, extrait de l'ouvrage "Designing Hypermedia for Learning"

[JONAS90], ainsi que par A. Tricot dans son article "Stratégies de navigation et stratégies d'apprentissage : pour l'approche expérimentale d'un problème cognitif" [TRICO93].

2.1.2. *Evaluation logicielle*

Les didacticiels hypermédias peuvent, d'après J. Nielsen, être évalués très classiquement grâce à des tests avec les utilisateurs. Le principal problème réside dans le fait que les bases de connaissances contiennent généralement un très grand nombre de modules et qu'il est donc impossible de les faire tous tester. Une manière de contourner le problème revient à suivre des guidelines lors du développement des modules et de compléter l'évaluation empirique d'évaluations heuristiques pour les modules non visités par les utilisateurs. Des exemples de ces guidelines sont donnés par [VANDERD94].

A noter qu'on peut utiliser le schéma d'évaluation continue des didacticiels présenté au point 1.2.2. pour vérifier l'ergonomie des modules.

Un soin particulier doit cependant être apporté à tout ce qui peut aider l'utilisateur à lutter contre la désorientation [TRICO93]. Les aides, intelligentes ou non, les index, les possibilités de retour en arrière se doivent d'être extrêmement clairs et bien étudiés. On a constaté en effet que des aides de ce type, si elles sont mal pensées, augmentent la charge cognitive des utilisateurs. Ces aides peuvent être évaluées par toute une série de guidelines, présentes dans la littérature, par exemple [GRISL96], [VANDERD94].

2.2.2. *Evaluation pédagogique*

En premier lieu, il convient de rappeler que l'évaluation pédagogique traditionnelle semble mise à mal par l'hypermédia. En effet, les objectifs ne peuvent être facilement définis si on laisse l'utilisateur choisir lui-même les sujets qui l'intéressent et qu'il peut se promener comme il l'entend à travers l'espace d'informations.

D'un autre côté, une liberté totale dans l'apprentissage est encore pour le moment du domaine de l'utopie. Au contraire, nous avons vu que de plus en plus d'auteurs plaident pour

des systèmes mixtes, c'est-à-dire où l'utilisateur est soumis à une "liberté contrôlée". D'ailleurs, la grosse majorité des systèmes existants sont basés sur cette optique. Une façon d'implémenter ces systèmes mixtes est de permettre aux étudiants de browser dans l'hyperdocument pour rechercher des informations sur un sujet précis. Par la suite, les élèves seront interrogés par le système sur la matière supposée apprise. La liberté s'applique donc dans la façon dont les étudiants effectuent leurs recherches, mais ces derniers restent néanmoins soumis à un encadrement dans la manière dont ils sont interrogés. Les problèmes relatifs à l'évaluation pédagogique perdent donc de leur complexité, puisqu'il devient possible de prédéfinir des objectifs d'apprentissage, comme en I.S.D. Une démarche semblable à celle utilisée classiquement dans les didacticiels peut donc être appliquée dans le cas de l'hypermédia.

Chapitre 8 :

Critique de l'évaluation du didacticiel hypermédia de langage C

Ainsi que nous l'avons vu au cours des chapitres précédents, l'évaluation d'un logiciel d'apprentissage, et plus particulièrement s'il s'agit d'un didacticiel hypermédia, n'est pas chose aisée. Le Cours Hypermédia de Langage C, mis au point par L. Toutain, a vu son évaluation ergonomique menée comme pour un logiciel classique, c'est-à-dire que nous nous sommes basés pour effectuer l'analyse de l'interface sur la méthode proposée par Jean Vanderdonck [VANDERD94].

1. Description de la méthode utilisée

Selon J. Vanderdonck, une analyse d'interface homme-machine se décompose en quatre étapes:

- l'analyse de la tâche et des utilisateurs,
- le logiciel incriminé,
- analyse fonctionnelle de l'interface,
- analyse ergonomique de l'interface.

1.1. Analyse de la tâche et des utilisateurs

Au cours de cette étape, on analyse les tâches réalisées par les utilisateurs. Il s'agit donc de mieux connaître la population cible du produit, ainsi que les actions que celle-ci effectue, et ceci sans le système d'information, car "la tâche elle-même en reste indépendante" [BODAR92]. Cette analyse se décompose en 4 étapes :

- l'analyse de la tâche proprement dite,
- l'identification et définition de la population en différents stéréotypes,
- la description ergonomique de la tâche,
- la description du poste de travail.

1.1.1. L'analyse de la tâche

Elle peut être réalisée de différentes manières. En général, on utilise fréquemment l'une ou l'autre de ces types d'analyse :

- l'analyse fonctionnelle, où l'on s'attache à mettre en évidence toutes les informations qui circulent lors de la réalisation des différentes tâches, ainsi que les traitements que vont subir ces informations à ce moment. Un schéma de la dynamique ou un diagramme de flux, par exemple, sont des outils efficaces pour mener à bien cette étape.
- l'analyse structurale, qui porte sur les objets de la tâche et sur ce que l'utilisateur peut faire sur ces derniers. Il s'agit donc d'effectuer une analyse orientée objet de la tâche.

1.1.2. Identification et définition de la population en différents stéréotypes

La population cible du logiciel doit être connue avec précision. Il s'agit tout d'abord de partitionner les utilisateurs selon leur profession et leur niveau d'expérience. Bien sûr, si lors du développement du logiciel une analyse fine des utilisateurs a déjà été menée, on peut s'en inspirer largement. Une fois les stéréotypes d'utilisateurs définis, chacun de ceux-ci est alors caractérisé selon les quatre paramètres décrits ci-dessous :

- l'expérience qu'ont les utilisateurs de la tâche à accomplir.
- l'expérience du système qui indique si l'utilisateur a déjà été confronté au système, que ce soit ou non dans l'accomplissement de sa tâche.
- la motivation de l'utilisateur est examinée par rapport à l'exécution de la tâche que celui-ci doit accomplir. La motivation peut avoir une influence considérable sur la façon dont le produit sera perçu. En effet, "la satisfaction des utilisateurs et l'utilisabilité du logiciel dépend pour une grande part de l'intérêt pour le contenu" [TRICO93, p.24].
- l'expérience d'un système d'interaction, qui indique si l'utilisateur a déjà été habitué à se servir d'un clavier étendu, d'une souris,...

1.1.3. Description ergonomique de la tâche

Cette description ne prend pas en compte le système d'information, mais se centre plutôt sur les stéréotypes et catégories d'utilisateurs identifiés ci-dessus, ainsi que sur la manière dont ces caractéristiques vont influencer l'accomplissement de la tâche. Les stéréotypes d'utilisateurs et la tâche sont caractérisés selon les 6 paramètres suivants :

- les pré-requis, qui indiquent les connaissances nécessaires à l'utilisateur afin que celui-ci puisse correctement effectuer sa tâche à l'aide du système informatique. En cas de pré-requis trop importants par rapport au niveau des utilisateurs, il faut mettre en place une session de formation.
- la productivité, qui représente "le nombre d'exécutions moyen de la tâche par unité de temps" [BODAR92].
- l'environnement objectif de la tâche, qui exprime la nécessité ou non pour l'utilisateur d'avoir à manipuler des objets extérieurs au système informatique pendant qu'il accomplit sa tâche, indépendamment de tout système. Il peut s'agir, par exemple, des documents papiers, des objets de bureau,...
- la structuration de la tâche, qui met en évidence le caractère dirigiste ou non du logiciel dans le cadre d'exécution de la tâche.
- l'importance de la tâche, qui indique les conséquences d'une exécution incorrecte de celle-ci. Il est évident que dans le cas où une erreur de l'utilisateur risque d'avoir des conséquences graves, l'interface doit être conçue pour limiter au maximum la survenance d'erreurs de ce type.
- la complexité de la tâche, qui met en lumière la difficulté relative d'une exécution correcte de celle-ci.

1.1.4. Description du poste de travail

Le poste de travail doit impérativement être pris en compte lors de l'évaluation du logiciel. En effet, l'environnement dans lequel le logiciel est utilisé peut avoir des conséquences importantes sur le bon déroulement de la tâche. Il faut donc tout d'abord "veiller à caractériser la disponibilité d'autres objets, documents, structures organisationnelles,

coopération avec d'autres programmes, utilisation conjointe avec un traitement de texte" [BODAR92]. Ensuite, il nous faudra qualifier par les deux paramètres type décrit ci-dessous:

- le type de traitement, qui permet de savoir si l'utilisateur, lorsqu'il effectue sa tâche au poste de travail ne peut effectuer que celle-ci (mono-traitement) ou si il peut en exécuter plusieurs autres au même moment (multi-traitement).
- la capacité de traitement, qui indique "le niveau d'interruptibilité, de parallélisme, de concurrence, d'interpénétrabilité des tâches effectuables au poste concerné" [BODAR92].

1.2 Le logiciel incriminé

Le logiciel à évaluer est analysé à cette étape, ainsi que ses "potentialités par rapport à la tâche" [BODAR92]. Plus précisément, on travaille sur :

- une description succincte du logiciel,
- une découpe du logiciel.

1.2.1. Une description succincte du logiciel

Les fonctionnalités du produit informatique sont énumérées et expliquées de manière concise à cette étape.

1.2.2. Une découpe du logiciel

Afin de pouvoir facilement évaluer le logiciel, celui-ci est découpé de deux manières différentes :

- découpe fonctionnelle, où l'on divise la tâche en un ensemble de sous-tâches que l'on enchaîne au moyen d'un schéma de la dynamique.
- découpe structurale, où l'on adopte une vue plus "objet", à la fois de la tâche et de l'interface. La tâche est vue comme un ensemble d'action à effectuer sur des objets et l'interface quant à elle est censée représenter électroniquement tous ces objets nécessaires à une bonne exécution de la tâche. Il faut donc à cette étape "faire porter la

comparaison sur les actions relatives à l'accomplissement de la tâche et les commandes proposées par l'interface" [BODAR92].

1.3. Analyse fonctionnelle de l'interface

Il s'agit dans ce chapitre de relever tous les types de composants de l'interface homme-machine. Pour ce faire, il convient de décrire les paramètres suivants :

- les moyens d'interaction utilisés,
- les objets interactifs utilisés,
- reconnaissance et listage du ou des styles d'interaction utilisés.

1.3.1. Les moyens d'interaction utilisés

Il s'agit ici de décrire tous le matériel à disposition de l'utilisateur pour interagir avec la machine, par exemple le clavier, la souris, l'écran tactile,...

1.3.2. Les objets interactifs utilisés

Un objet interactif est la représentation dans l'interface d'une information ou d'une action déclenchable par l'utilisateur. Puisque ces objets interactifs sont à la base de la communication entre l'utilisateur et le logiciel, il est important lors d'une évaluation de les répertorier, avant de les rattacher à telle ou telle catégorie décrite ci-dessous :

- les objets d'action, c'est-à-dire tout ce qui concerne les menus,
- les objets de défilement c'est-à-dire tout ce qui permet à l'utilisateur de se déplacer à l'intérieur de l'interface, par exemple à travers des pages, des écrans,
- les objets statiques qui n'ont qu'un pur objectif informatif,
- les objets de contrôle que l'utilisateur peut "manipuler pour introduire des données ou déclencher une action" [BODAR92],
- objets de feed-back, qui informent l'utilisateur des conséquences de ses actions sur l'interface,

- les objets de dialogue, c'est-à-dire tout ce qui concerne les fenêtres, boîtes de dialogue et panneaux de contrôle.

1.3.3. Reconnaissance et listage du ou des styles d'interaction utilisés

Ce paramètre décrit la manière dont l'interface présente les informations à l'utilisateur et comment ce dernier peut à son tour entrer des données dans la machine. Les styles d'interaction possibles sont :

- le langage naturel,
- le langage de commande,
- le langage d'interrogation,
- les questions/réponses,
- les touches de fonction,
- la sélection de menu,
- le remplissage de forme,
- le multi-fenêtrage,
- la manipulation directe
- l'interaction iconique.

Le style d'interaction optimal par rapport à une interface donnée dépend fortement des caractéristiques identifiées lors des étapes précédentes. Dans notre cas, il est évident que le style d'interaction utilisé principalement sera le remplissage de forme, puisque le produit consiste en un didacticiel avec tests du type questions/réponses.

1.4. L'analyse ergonomique de l'interface.

J. Vanderdonck préconise dans cette étape de la méthode l'utilisation des évaluations heuristiques pour mener à bien l'évaluation ergonomique proprement dite. C'est la démarche qui a été adoptée pour réaliser l'évaluation de l'interface du Cours Hypermédia de Langage C.

2. Critique de la méthode utilisée

La méthodologie décrite ci-dessus et employée pour évaluer le Cours de Hypermédia de Langage C, a des avantages et des défauts. Elle nous a paru sur le moment assez bien adaptée aux exigences de la situation. Quoiqu'il en soit, il semble important aujourd'hui de reconsidérer d'un oeil critique le travail effectué, et ce principalement sur les 4 points suivants:

- analyse des utilisateurs, de la tâche et du poste de travail,
- évaluation heuristique de l'ergonomie de l'interface,
- absence d'analyse de la documentation et du manuel d'utilisation,
- prise en compte des aspects hypermédia du produit.

2.1. Analyse des utilisateurs, de la tâche et du poste de travail

La méthode de J. Vanderdonck possède le gros avantage d'intégrer des étapes d'analyse précise de la tâche, des utilisateurs et du poste de travail. Elle m'a donc donc sembler particulièrement intéressante dans le cadre de l'évaluation du Cours Hypermédia de Langage C, où ces aspects ne figuraient explicitement nulle part, aucun document de développement du produit n'étant disponible.

Or, dans la littérature, ces aspects sont considérés comme extrêmement importants. Toutes les méthodologies de développement, aussi bien de logiciels que de didacticiels, insistent en effet sur la nécessité de bien cerner les utilisateurs, leurs attentes quant au produit, les tâches qu'ils devront accomplir à l'aide du système et l'environnement dans lequel s'effectuera ce travail. Dans l'évaluation, les étudiants ont été classiquement réparti selon leur section et leurs connaissances préalables du langage C. Peut-être aurait-il été intéressant de davantage insister sur ce qu'ils espéraient du logiciel, quels aspects ils auraient souhaiter voir développer,... Mais l'évaluation se déroulant à posteriori, nous avons pensé que ces considérations étaient à ce moment d'un impact réduit. Par ailleurs, la tâche et le poste de travail ont été analysés selon la méthode décrite.

2.2. Evaluation heuristique de l'ergonomie de l'interface

Cette technique, en préconisant pour juger de l'ergonomie la méthode heuristique, ne semble pas très adaptée à l'évaluation finale d'un produit, d'autant plus que celui-ci avait déjà été diffusé avant que l'inspection de l'interface ne commence. Il aurait sans doute à ce moment mieux valu mener une évaluation empirique, quoique les élèves de cette année, ayant déjà tous eu affaire au didacticiel, ne pouvaient plus être considérés comme des utilisateurs parfaitement représentatifs de la population estudiantine future. En conséquence, l'évaluation empirique s'avérant difficile à mettre en oeuvre, la méthodologie heuristique a donc été choisie par défaut.

Un argument qui jouait d'ailleurs en faveur de l'évaluation empirique, mais dont nous n'avons véritablement pris conscience que plus tard, vient du nombre de personnes à impliquer dans l'inspection de l'interface. En effet, la méthode heuristique requiert l'avis d'une équipe de plusieurs personnes pour aboutir à une évaluation assez complète et à des propositions de correction conséquentes. En travaillant seul, on court donc le risque de passer à côté d'un certain nombre d'erreurs ergonomiques. Une façon de contourner le problème, si on peut dire, a été de compléter l'évaluation heuristique d'un questionnaire permettant de mieux comprendre la façon dont les étudiants voyaient le produit, ainsi que leurs principaux griefs, remarques,... quant à son utilisation. De plus, avant de commencer à analyser l'interface, nous avons décidé de suivre le cours de C, de faire les exercices proposés,... ce qui a amélioré notre compréhension de la tâche que devaient mener les étudiants. Cela nous a également sensibilisés davantage aux problèmes rencontrés par les utilisateurs.

2.3. Absence d'analyse de la documentation et du manuel d'utilisation

Un reproche que l'on pourrait faire à la méthode utilisée est qu'elle ne se préoccupe pas assez de la documentation et des aspects 'hors-logiciel' tels le manuel d'utilisation, la formation à assurer aux opérateurs,... Cet aspect n'est cependant pas trop gênant dans ce cas précis. En effet, le didacticiel considéré n'était pas accompagné de ce type de documents. Par conséquent, le fait que la méthodologie ne prescrive pas d'analyse du manuel n'a pas ici porté à conséquence.

Une question qu'on peut se poser relève de l'opportunité d'intégrer à ce cours un manuel d'utilisation ou de la documentation ad hoc. Il est vrai que ce type de support papier peut toujours être intéressant pour aider à dépanner un utilisateur. Mais la structuration importante du cours (ce qui implique un risque très faible de se perdre) et le peu de difficulté manifesté par les étudiants pour se servir du didacticiel ont été déterminants dans la décision de ne pas intégrer un supplément papier au produit.

2.4. Prise en compte des aspects hypermédia du produit

La méthode préconisée ne tient pas explicitement compte du support employé. Les caractéristiques de l'hypermédia et leurs conséquences sur l'évaluation du didacticiel n'apparaissent donc pas clairement. Cependant, le Cours Hypermédia de Langage C ayant une structure très particulière, l'analyse du didacticiel peut se réaliser de manière assez classique. En effet, les possibilités du support sont relativement sous-exploitées dans ce produit. Le cours suit un déroulement quasi linéaire, l'étudiant n'ayant que peu de possibilité de browser à travers l'hyperdocument. La désorientation, si redoutée dans les environnements hypermédia, a donc vu sa probabilité d'occurrence extrêmement réduite.

Il aurait été intéressant de permettre aux étudiants de pouvoir répondre aux questions de tests en recherchant librement dans l'hyperdocument les réponses. Comme nous l'avons dit, ce cours sous-exploite les possibilités du support. Nous n'avons pas laissé aux élèves cette possibilité. Dans le Cours Hypermédia de Langage C, en effet, on présente à l'utilisateur une table des matières, lui indiquant les chapitres auxquels il peut avoir accès. Il ne peut dès lors se déplacer librement que dans les modules que lui indique le système. On pourrait reprocher ce manque de liberté laissé à l'utilisateur. Mais d'un autre côté, la charge cognitive est moins élevée de cette façon et le risque de se perdre, extrêmement réduit. En résumant grossièrement, nous dirons qu'il a fallu faire un choix entre liberté et facilité d'utilisation. Pour notre part, nous avons opté pour la deuxième solution.

Conclusions

L'évaluation d'un didacticiel est une tâche complexe car elle nécessite des connaissances dans des domaines aussi divers que la pédagogie, l'ergonomie et l'informatique. Les méthodologies de développement de didacticiels sont d'ailleurs soumises en général aux influences en provenance du génie logiciel. Or, au cours de ces dernières années, une évolution s'est fait jour dans le domaine de la conception de logiciels, évolution qui implique de plus en plus la prise en compte des utilisateurs et de l'ergonomie de l'interface. Ces objectifs ne peuvent cependant être remplis en déans des coûts raisonnables qu'en intégrant tout au long du cycle de développement une approche évaluative continue.

Cette nouvelle approche du développement d'un produit informatique ne va cependant pas de soi, surtout dans le cadre de la mise au point d'un didacticiel, où traditionnellement, on s'est toujours centré davantage sur les évaluations pédagogiques qu'ergonomiques. Il est donc important de mettre au point des méthodologies de conception des logiciels d'apprentissage qui intègrent le plus harmonieusement possible les différentes méthodes d'évaluation décrites en génie logiciel.

Nous espérons que ce mémoire aborde de façon suffisamment précise ces diverses dimensions afin de fournir une base solide à nos éventuels successeurs. Nous avons pour cela préféré progresser pas à pas dans les chapitres, en commençant par rappeler en quoi consiste l'enseignement assisté par ordinateur et en décrivant en détail le courant de pensée dominant dans cette discipline. Nous avons ensuite brièvement abordé un nouveau support à l'E.A.O., l'hypermédia, avant de passer à la problématique de l'évaluation proprement dite. Pour ce faire, nous avons passé en revue les techniques existant en génie logiciel et leur intégration dans le cycle de développement, pour finalement examiner comment ces techniques pouvaient s'intégrer lors de la mise au point d'un didacticiel. Nous avons terminé en soulignant les difficultés d'évaluation à-posteriori d'un didacticiel hypermédia, en décrivant le travail effectué lors du stage de fin d'études.

La méthodologie proposée, en intégrant l'optique de l'évaluation ergonomique continue lors du développement des didacticiels, permet de résoudre le problème délicat du choix des techniques à employer et à quel moment pour s'assurer de la qualité des interfaces. Or, on sait

que cette qualité a une influence certaine sur la facilité d'apprentissage des étudiants. Encore faut-il que cette méthodologie soit appliquée par les différentes équipes de projet travaillant à la mise au point de logiciels éducatifs. En effet, cette manière de procéder est assez théorique et doit encore être validée et affinée par les milieux professionnels. Un aspect important pour l'amélioration de l'ergonomie consiste sans nul doute dans la mise en place rapide des différentes techniques d'évaluation automatique. Les recherches dans cette voie progressent et il serait alors utile d'essayer d'intégrer dans le cycle de développement la prise en compte de ces évaluations.

Par ailleurs, ce travail a été réalisé dans une optique informatique, la manière de réaliser l'évaluation pédagogique n'ayant été qu'esquissée. Il serait intéressant de creuser davantage dans cette voie, afin de parvenir à une évaluation véritablement complète de tous les aspects d'un didacticiel.

Bibliographie

- [BARON93] BARON G.-L., BAUDE J., de LA PASSARDIERE B. (1993), *Hypermédiat et Apprentissages 2, Actes des deuxièmes journées scientifiques Lille, 24-25 mars 1993*, Institut National de Recherche Pédagogique, Paris
- [BESN88] BESNAINOU R., MULLER C., THOUIN T. (1988), *Concevoir et Utiliser un didacticiel*, Les Editions d'Organisation, Paris.
- [BIAS94] BIAS R.(1994), *The Pluralistic Usability Walkthrough: Coordinated Empathies*,
in MACK R., NIELSEN J. (1994), *Usability Inspection Methods*, A.P. Professional, Boston.
- [BODART92] BODART F.(1992), *Interface Homme/Machine. Cours de seconde licence et maîtrise en Informatique*, Institut d'Informatique, Facultés Universitaires Notre-Dame de la Paix, Namur.
- [BRADF94] BRADFORD J.(1994), *Evaluating High-Level Design Synergistic Use of inspection and Usability Method for Evaluating Early Software Design*,
in MACK R., NIELSEN J. (1994), *Usability Inspection Methods*, A.P. Professional, Boston.
- [DEPOV91] DEPOVER C., QUINTIN J.-J., DE LIEVRE B. (1991), *Eléments pour un modèle pédagogique adapté aux possibilités d'un environnement hypermédia*,
in de LA PASSARDIERE B., BARON G.-L. (1991), *Hypermédiat et Apprentissages , Actes des premières journées scientifiques, 24-25 septembre 1991*, Institut National de Recherche Pédagogique, Paris

- [DESURV94] DESURVIRE H. (1994), *Faster, Cheaper!! Are Usability Inspection Methods as Effective as Empirical Testing?*,
in MACK R., NIELSEN J. (1994), *Usability Inspection Methods*, A.P. Professional, Boston.
- [DUFRES91] DUFRESNE A. (1991), *Ergonomie cognitive, Hypermédias et Apprentissages*,
in de LA PASSARDIERE B., BARON G.-L. (1991), *Hypermédias et Apprentissages , Actes des premières journées scientifiques, 24-25 septembre 1991*, Institut National de Recherche Pédagogique, Paris
- [GOBER95] GOBERT X. (1995), *Contribution à l'Enseignement Assisté par Ordinateur dans le domaine des télécommunications. Un didacticiel pour Internet*,
Institut d'informatique, Facultés Universitaires Notre-Dame de la Paix, Namur
- [GRAB90] GRABINGER R., JONASSEN D. (1990), *Problem and Issues in designing Hypertext/Hypermedia for Learning*,
in JONASSEN D, MANDL H. (1990), *Designing Hypermedia for Learning*, Springer, Berlin.
- [GRISL96] GRISLIN M., KOLSKI C. (1996), *Evaluation des interfaces homme-machine lors du développement des systèmes interactifs*,
in *Technique et Science Informatiques*, n°15, Hermes, Paris.
- [JONAS90] JONASSEN D, MANDL H. (1990), *Designing Hypermedia for Learning*, Springer, Berlin.
- [KAHN94] KAHN M., PRAIL A. (1994), *Formal Usability Inspections*,
in MACK R., NIELSEN J. (1994), *Usability Inspection Methods*, A.P. Professional, Boston.

- [KARAT94] KARAT C. (1994), *A Comparison of User Interface Evaluation Methods*,
in MACK R., NIELSEN J. (1994), *Usability Inspection Methods*, A.P. Professional, Boston.
- [KEYSE89] DE KEYSER V., VAN DAELE A.(1989), *L'ergonomie de conception*, De Boeck-Wesmael, Bruxelles
- [LANDO90] LANDOW G. (1990), *Popular Fallacies About Hypertext*,
in JONASSEN D, MANDL H. (1990), *Designing Hypermedia for Learning*, Springer, Berlin.
- [LECLE77] LECLERCQ D., DONNAY J., DE BAL R. (1977),
Construire un cours programmé, Nathan, Paris.
- [LOGAN82] LOGAN (1982), *Instructional System Development, An Internal View of Theory and Practice*, Academic Press, New-York.
- [MACK94] MACK R., NIELSEN J. (1994), *Usability Inspection Methods*, A.P. Professional, Boston.
- [MARCH90] MARCHIONINI G. (1990), *Evaluating Hypermedia-Based Learning*,
in JONASSEN D, MANDL H. (1990), *Designing Hypermedia for Learning*, Springer, Berlin.
- [MAYES90] MAYES T., KIBBY M., ANDERSON T. (1990),
Learning About Learning From Hypertext
in JONASSEN D, MANDL H. (1990), *Designing Hypermedia for Learning*, Springer, Berlin.

- [MAYES93] MAYEST T. (1993), *Hypermédiats et outils cognitifs*,
in BARON G.-L., BAUDE J., de LA PASSARDIERE B.
(1993), *Hypermédiats et Apprentissages 2, Actes des
deuxièmes journées scientifiques Lille, 24-25 mars 1993*,
Institut National de Recherche Pédagogique, Paris
- [MEDA90] MEDA J. (1990), *Evaluer les logiciels de Formation*,
Les Editions d'Organisation, Paris.
- [NOIRH95] NOIRHOMME FRAITURE M., GOFFINET L. (1995),
*Multimédia, Actes de la journée d'information sur le
multimédia*, Les Presses Universitaires de Namur,
Namur.
- [NIELS89] NIELSEN J. (1989), *Hypertexte et Hypermédia*,
Academic Press Inc., San Diego.
- [NIELS93] NIELSEN J. (1993), *Usability engineering*, A.P.
Professional, Boston.
- [NIELS94] NIELSEN J. (1994), *Heuristic Evaluation*,
in MACK R., NIELSEN J. (1994), *Usability Inspection
Methods*, A.P. Professional, Boston.
- [NOTAIS95] NOTAISE J., DUSANTER O., BARDA J. (1995),
*Dictionnaire du multimédia : audiovisuel, informatique,
télécommunications*, AFNOR, Paris.
- [PASSAR91] de LA PASSARDIERE B., BARON G.-L. (1991),
*Hypermédiats et Apprentissages , Actes des premières
journées scientifiques, 24-25 septembre 1991*, Institut
National de Recherche Pédagogique, Paris

- [RHEA91] RHEAUME J. (1991), *Hypermédiat et stratégies pédagogiques*,
in de LA PASSARDIERE B., BARON G.-L. (1991),
Hypermédiat et Apprentissages, Actes des premières
journées scientifiques, 24-25 septembre 1991, Institut
National de Recherche Pédagogique, Paris
- [ROMIS90] ROMISZOWSKI A. (1990), *The Hypertext/Hypermedia
solution-But What Exactly is the Problem?*,
in JONASSEN D, MANDL H. (1990), *Designing
Hypermedia for Learning*, Springer, Berlin.
- [RUBIN94] RUBIN J. (1994), *Handbook of Usability testing : How
to plan, design and conduct effective tests*, John Wiley
& Sons Inc., New-York
- [TRICO93] TRICOT A. (1993), *Stratégies de navigation et
stratégies d'apprentissage : pour l'approche
expérimentale d'un problème cognitif*,
in BARON G.-L., BAUDE J., de LA PASSARDIERE B.
(1993), *Hypermédiat et Apprentissages 2, Actes des
deuxièmes journées scientifiques Lille, 24-25 mars 1993*,
Institut National de Recherche Pédagogique, Paris
- [VANDERD94] VANDERDONCKT J. (1994), *Guide Ergonomique des
Interfaces Homme/Machine*, Les Presses Universitaires
de Namur, Namur.
- [WHART94] WHARTON C., RIEMAN J., LEWIS C., POLSON P.
(1994), *The Cognitive Walkthrough Method : A
Practitioner's Guide*,
in MACK R., NIELSEN J. (1994), *Usability Inspection
Methods*, A.P. Professional, Boston.

[WIXON94]

WIXON D., JONES S., TSE L., CASADAY G. (1994),
*Inspections and design reviews : Framework, History,
and reflection*

in MACK R., NIELSEN J. (1994), *Usability Inspection
Methods*, A.P. Professional, Boston.