



**HAL**  
open science

# Spécification et conception d'un environnement d'aide à l'inspection des Systèmes Interactifs, basé sur la méthode Cognitive Walkthrough

Thomas Mahatody

► **To cite this version:**

Thomas Mahatody. Spécification et conception d'un environnement d'aide à l'inspection des Systèmes Interactifs, basé sur la méthode Cognitive Walkthrough. Interface homme-machine [cs.HC]. 'Université de Valenciennes et du Hainaut-Cambrésis, 2010. Français. NNT : 2010VALE0024 . tel-01274648

**HAL Id: tel-01274648**

**<https://hal.science/tel-01274648>**

Submitted on 17 Feb 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE

pour l'obtention du

## Doctorat de l'Université de Valenciennes et du Hainaut-Cambrésis

*Spécialité Automatique et Informatique des Systèmes Industriels et Humains  
Mention Informatique*

### Thomas MAHATODY

Ingénieur ENI

## Spécification et conception d'un environnement d'aide à l'inspection des Systèmes Interactifs, basé sur la méthode Cognitive Walkthrough

Soutenue publiquement le 8 Juillet 2010 devant le jury composé de

M. Luigi LANCIERI	Professeur à l'Université de Lille 1	Président
M. Ahmed SEFFAH	Professeur à l'Ecole Hôtelière de Lausanne	Rapporteur
M. Bertrand DAVID	Professeur à l'Ecole Centrale de Lyon	Rapporteur
M. Christian BASTIEN	Maître de Conférences HDR à l'Université de Metz	Examineur
M. Christophe KOLSKI	Professeur à l'Université de Valenciennes	Directeur
M. Mouldi SAGAR	Maître de conférences à l'Université de Valenciennes	Co-Encadrant

# Avant propos

Cette thèse est le résultat de quatre années de recherche et d'effort. Dans les grandes choses, avant l'effort qui réussit, il y a presque toujours des efforts qui passent inaperçus. [*Laure Conan, Extrait de l'oublié*].

Pour les efforts qu'ils ont fait, je tiens à remercier :

Bertrand DAVID et Ahmed SEFFAH pour avoir évalué ce travail. Leurs remarques et leur patience m'ont permis d'améliorer sensiblement la présentation de mon travail,

Christophe KOLSKI et Mouldi SAGAR, pour m'avoir accueilli, guidé, encadré pendant ces quatre années. J'ai apprécié leurs conseils et leurs encouragements qui ont permis la concrétisation de cette thèse,

Luigi LANCIERI et Christian BASTIEN d'avoir accepté de participer à mon jury de thèse,

tous mes entourages professionnels au LAMIH et à l'Université de Fianarantsoa,

ma famille pour m'avoir soutenu et encouragé dans cette aventure,

mon épouse Harivelo et mes chers enfants Douglas, Christina, Kevin et Shania pour avoir accepté de se séparer de moi pendant quatre ans. En guise de reconnaissance, je vous dédie cette thèse.

à Harivelo, Douglas, Christina, Kevin, Shania

## Table des matières

Avant propos.....	2
Liste des figures.....	9
Liste des tableaux.....	11
Introduction générale.....	12
Chapitre 1- Eléments clés liés à la prise en compte des facteurs humains dans la conception et l'évaluation d'IHM.....	15
1.1- Introduction.....	16
1.2- Les artefacts.....	17
1.2.1 - La conception centrée utilisateur.....	17
1.2.2 - Modèle de l'artefact.....	18
1.3- Les tâches et leur modélisation.....	19
1.3.1 - Hierarchical Task Analysis.....	20
1.3.2 - Groupware Task Analysis (GTA).....	21
1.3.3 - Méthode Analytique de Description de tâches (MAD).....	22
1.3.4 - CTT.....	23
1.3.5 - Modèle de description de l'activité (N-MDA).....	25
1.4- Les utilisateurs et leur modélisation.....	26
1.4.1 - La loi de Fitts.....	26
1.4.2 - Le modèle du processeur humain.....	26
1.4.3 - GOMS.....	27
1.4.4 - CCT.....	27
1.4.5 - Le modèle de Rasmussen.....	28
1.4.6 - Le modèle de Norman.....	29
1.4.7 - Le modèle de Kintsch.....	29
1.4.8 - Le modèle EXPL.....	30
1.4.8.1 - L'heuristique de l'identité.....	30
1.4.8.2 - L'heuristique des extrémités lâches.....	30
1.4.8.3 - L'heuristique de l'action précédente.....	30
1.4.9 - Le modèle CE+.....	30
1.4.10 - Le modèle LICAI.....	31
1.4.11 - Le modèle CoLiDeS.....	31
1.4.12 - La conscience de la situation et la représentation occurrente.....	32
1.4.13 - Modélisation de l'utilisateur.....	32
1.5- Règles ergonomiques.....	34
1.5.1 - Généralités.....	34
1.5.2 - Normes liées à l'IHM.....	35
1.5.2.1 - Qualité d'utilisation.....	35
1.5.2.2 - Qualité de l'artefact.....	35
1.5.2.3 - Normes pour la spécification des interfaces homme-machine.....	36
1.5.2.4 - Normes pour la conception des interfaces homme-machine.....	37
1.5.2.5 - Normes relatives aux critères d'évaluation des interfaces homme-machine.....	38
1.5.2.6 - Normes relatives à la qualité du processus de développement.....	38
1.5.2.7 - Normes relatives à la qualité de l'organisation.....	38
1.6- Evaluation des systèmes interactifs : principes et critères.....	39
1.6.1 - Les heuristiques de Nielsen.....	40
1.6.2 - Les critères ergonomiques de Bastien et Scapin.....	41
1.6.3 - Méthodes d'évaluation des systèmes interactifs.....	44

1.6.3.1 - Les tests d'utilisabilité.....	44
1.6.3.2 - Les méthodes d'inspection de l'utilisabilité.....	45
1.7- Conclusion.....	46
Chapitre 2- La méthode Cognitive Walkthrough et ses différentes versions et variantes.....	47
2.1- Introduction.....	48
2.2- Evolution des versions originales.....	49
2.2.1 - Première version.....	49
2.2.2 - Deuxième version.....	49
2.2.3 - Troisième version.....	50
2.3- Les autres variantes.....	51
2.3.1 - Heuristic Walkthrough (HW).....	51
2.3.2 - Norman's Cognitive Walkthrough.....	51
2.3.3 - Streamlined Cognitive Walkthrough.....	53
2.3.4 - Cognitive Walkthrough for the Web.....	53
2.3.5 - Groupware Walkthrough.....	55
2.3.6 - Activity Walkthrough.....	56
2.3.7 - Interaction Walkthrough.....	56
2.3.8 - Cognitive Walkthrough with users.....	58
2.3.9 - Extended Cognitive Walkthrough.....	58
2.3.10 - Distributed Cognitive Walkthrough.....	59
2.3.11 - Enhanced Cognitive Walkthrough.....	60
2.4- Synthèses.....	61
2.4.1 - Synthèses selon les aspects conceptuels, méthodologiques et technologiques.....	61
2.4.2 - Synthèse sur les études comparatives.....	67
2.4.2.1 - Etude de Lewis et al. [1990].....	67
2.4.2.2 - Etude de Polson et al. [1992].....	67
2.4.2.3 - Etude de Sears [1997].....	68
2.4.2.4 - Etude de Granoller et Lorès [2004].....	68
2.4.2.5 - Etude de Kato et Hori [2005].....	68
2.4.2.6 - L'étude d'Eden [2008].....	69
2.4.2.7 - Etude de Jeffries et al. [1991].....	69
2.4.2.8 - Etude de John et Marks [1997].....	70
2.4.3 - Synthèses sur les études non comparatives.....	77
2.4.3.1 - Etude de Rizzo et al. [1997].....	77
2.4.3.2 - Etude de Spencer [2000].....	77
2.4.3.3 - Etude de Blackmon et al. [2002].....	77
2.4.3.4 - Etude de Pinelle et Gutwin [2002].....	78
2.4.3.5 - Etude de Huart et al. [2004].....	78
2.4.3.6 - Etude de Bertelsen [2004].....	79
2.4.3.7 - Etude de Ryu et Monk [2004].....	79
2.4.3.8 - Etude de Bligard et Osvalder [2007].....	79
2.4.4 - Synthèse en vue d'une première aide au choix d'une version ou variante.....	82
2.5- Conclusion.....	84
Chapitre 3- Spécification d'un environnement d'évaluation de systèmes interactifs basée sur la méthode Cognitive Walkthrough.....	86
3.1- Introduction.....	87
3.2- Le processus d'évaluation.....	87
3.2.1 - Les acteurs de l'évaluation.....	88

3.2.2 - Les étapes de l'évaluation.....	88
3.2.2.1 - La phase de préparation.....	88
3.2.2.1.1 Spécification du système à évaluer.....	88
3.2.2.1.2 Modélisation des utilisateurs finaux.....	89
3.2.2.1.3 Choix des scénarii à évaluer.....	89
3.2.2.1.4 Analyse des tâches.....	90
3.2.2.1.5 Choix d'une méthode d'évaluation.....	90
3.2.2.2 - La phase d'évaluation.....	90
3.2.2.2.1 Prise en main de l'évaluation.....	90
3.2.2.2.2 Exécution de l'évaluation.....	90
3.2.2.3 - La phase d'analyse des résultats.....	90
3.3- Modélisation statique de l'évaluation.....	91
3.3.1 - Le modèle de l'artefact.....	91
3.3.2 - Le modèle de l'utilisateur.....	93
3.3.3 - Le modèle de tâche.....	94
3.3.3.1 - Scénario.....	94
3.3.3.2 - Tâche.....	94
3.3.4 - Le modèle des méthodes.....	96
3.3.4.1 - Représentation de la première version de Lewis et al. [1990] (CW1).....	98
3.3.4.2 - Représentation de la troisième version de Warton et al. [1994] (CW3).....	99
3.3.4.3 - Représentation de Heuristic Walkthrough (HW).....	99
3.3.4.4 - Représentation de Norton CW.....	100
3.3.4.5 - Représentation de Streamlined CW.....	101
3.3.4.6 - Représentation de CWW.....	101
3.3.4.7 - Représentation de Extended CW.....	102
3.3.4.8 - Représentation de Enhanced CW.....	103
3.3.4.9 - Représentation de Distributed CW.....	104
3.3.4.10 - Représentation de la méthode d'évaluation heuristique de Nielsen.....	105
3.3.4.11 - Représentation de Groupware Walkthrough (GW).....	106
3.3.5 - Le modèle de résultat.....	106
3.4- Modélisation dynamique de l'évaluation.....	107
3.4.1 - Spécification de l'artefact.....	107
3.4.1.1 - Description du système.....	108
3.4.1.2 - Constitution des lexiques.....	109
3.4.2 - Modélisation de l'utilisateur.....	110
3.4.2.1 - Identification de l'utilisateur.....	110
3.4.2.2 - Description de la connaissance en informatique de l'utilisateur.....	111
3.4.2.3 - Description de la connaissance de l'utilisateur sur l'artefact.....	111
3.4.3 - Choix de scénarii à évaluer.....	112
3.4.4 - Analyse de tâches.....	113
3.4.5 - Choix de méthode d'évaluation.....	114
3.4.6 - Prise en main de l'évaluation.....	115
3.4.7 - Identification de l'évaluateur.....	116
3.4.8 - Inspection.....	117
3.4.9 - Analyse des résultats.....	118
3.5- Conclusion.....	119
Chapitre 4- Maquettage de CWE, environnement d'aide à l'évaluation de systèmes interactifs, basé sur la méthode Cognitive Walkthrough.....	120

4.1- Introduction.....	121
4.2- Objectifs architecturaux et contraintes.....	121
4.2.1 - Exigences logicielles.....	121
4.2.2 - Exigences spéciales de développement.....	121
4.3- Architecture de CWE.....	123
4.3.1 - Décisions architecturales.....	123
4.3.2 - Vue cas d'utilisation.....	125
4.3.2.1 - Cas d'utilisation : « Décrire le système à évaluer ».....	126
4.3.2.2 - Cas d'utilisation : « Constituer les lexiques ».....	126
4.3.2.3 - Cas d'utilisation : « Identifier les utilisateurs ».....	127
4.3.2.4 - Définir les connaissances informatiques des utilisateurs.....	128
4.3.2.5 - Cas d'utilisation : « Décrire les connaissances des utilisateurs concernant le système ».....	129
4.3.2.6 - Cas d'utilisation : « Spécifier le profil de l'évaluateur ».....	130
4.3.2.7 - Cas d'utilisation : « Choisir les scénarii à évaluer ».....	133
4.3.2.8 - Cas d'utilisation : « Analyser les tâches ».....	134
4.3.2.9 - Cas d'utilisation : « Choisir les méthodes d'évaluation ».....	135
4.3.2.10 - Cas d'utilisation : « Inspecter les tâches ».....	136
4.3.2.11 - Cas d'utilisation : « Calculer les tableaux de bord ».....	137
4.3.3 - Vue logique.....	139
4.3.3.1 - Paquetages de conception architecturale significatifs.....	139
4.3.3.2 - Illustrations de réalisation des cas d'utilisation.....	142
4.4- Conclusion.....	150
Chapitre 5- Evaluation de la première maquette de l'environnement CWE.....	151
5.1- Introduction.....	152
5.1.1 - La page de titre.....	152
5.1.2 - Un sommaire exécutif.....	152
5.1.3 - Une introduction.....	153
5.1.3.1 - Description complète du produit.....	153
5.1.3.2 - Objectifs de l'évaluation.....	153
5.1.4 - La méthodologie.....	153
5.1.5 - Résultats.....	154
5.1.5.1 - Analyse des données.....	154
5.1.5.2 - Présentation des résultats.....	154
5.2- Sommaire exécutif.....	154
5.3- Présentation de l'évaluation.....	154
5.3.1 - Description complète du produit à évaluer.....	154
5.3.2 - Objectifs de l'évaluation.....	155
5.4- Méthode.....	156
5.4.1 - Participants.....	156
5.4.2 - Contexte de produit évalué.....	158
5.4.2.1 - Tâches.....	158
5.4.2.2 - Environnement d'évaluation.....	158
5.4.2.3 - Configuration des ordinateurs des participants.....	159
5.4.3 - Stratégie de l'évaluation.....	159
5.4.3.1 - Procédure d'évaluation.....	159
5.4.3.2 - Instructions générales aux participants.....	160
5.4.3.3 - Instructions concernant les tâches des participants.....	160



5.4.4 - Métriques utilisées.....	160
5.5- Résultats de l'évaluation.....	161
5.5.1 - Résultats des tests fonctionnels.....	161
5.5.2 - Résultats de Cognitive Walkthrough version 3.....	163
5.5.3 - Discussion sur les deux évaluations.....	165
5.6- Conclusion.....	166
Conclusion et perspectives.....	167
Bibliographie.....	170
Annexes.....	181
Annexe A- Plan d'évaluation.....	181
Annexe B- Cahier d'évaluation Test fonctionnel.....	185
Annexe C- Cahier d'évaluation Cognitive Walkthrough.....	202
Annexe D- Fiche d'évaluation.....	211
Annexe E- Fiche problèmes.....	212

## Liste des figures

Figure 1.1 : Pyramide artefact - utilisateur - tâche - environnement.....	16
Figure 1.2 : Exemple de diagramme HTA.....	21
Figure 1.3 : Exemple de diagramme avec Groupware Task Analysis.....	21
Figure 1.4 : Exemple de diagramme avec CTTE.....	25
Figure 1.5 : Exemple de diagramme avec K-MAD.....	25
Figure 1.6 : Processus de conception centrée utilisateur.....	39
Figure 1.7 : Décomposition de Nielsen [1993].....	40
Figure 2.1 : Le modèle de modifié de Norman adapté de [Rizzo et al., 1997].....	52
Figure 2.2 Les étapes de CW for the Web (adapté de [Blackmon et al., 2002]).....	54
Figure 2.3 Théorie de l'interaction cyclique [Monk, 1998].....	57
Figure 2.4 : Modèle de Norman étendu adapté de [Kato et Hori, 2005].....	58
Figure 3.1 : Diagramme de cas d'utilisation.....	87
Figure 3.2 : Modèle de domaine.....	92
Figure 3.3 : Spécification du système à évaluer.....	108
Figure 3.4 : Constitution des lexiques.....	109
Figure 3.5 : Identification de l'utilisateur.....	111
Figure 3.6 : Description de la connaissance en informatique de l'utilisateur.....	111
Figure 3.7 : Description de la connaissance de l'utilisateur sur l'artefact.....	112
Figure 3.8 : Choix de scenarii à évaluer.....	112
Figure 3.9 : Analyse de tâches.....	114
Figure 3.10 : Apprentissage des différentes méthodes.....	115
Figure 3.11 : Apprentissage de CWE.....	115
Figure 3.12 : Appropriation des entrées de l'évaluation.....	116
Figure 3.13 : Identification de l'évaluateur.....	116
Figure 3.14 : Inspection de tâches.....	117
Figure 4.1 : Les "4+1" vues de Kruchten [1995].....	122
Figure 4.2 : cweMYRS.properties (extrait).....	123
Figure 4.3 : cweMYRS_en.properties (extrait).....	123
Figure 4.4 : Fonctionnement du service de persistance.....	124
Figure 4.5 : Correspondance entre association de classes et association d'entité relation.....	125
Figure 4.6 : Cas d'utilisation significatifs.....	125
Figure 4.7 : Onglet de description du système à évaluer.....	126
Figure 4.8 : Onglet de constitution des lexiques.....	127
Figure 4.9 : Onglet d'identification de l'utilisateur.....	128
Figure 4.10 : Onglet de description des connaissances en informatique de l'utilisateur.....	129
Figure 4.11 : Onglet de description des connaissances des utilisateurs concernant le système.....	130
Figure 4.12 : Onglet d'identification de l'évaluateur.....	131
Figure 4.13 : Onglet de description de l'expérience en informatique de l'évaluateur.....	131
Figure 4.14 : Onglet de description des expériences en Cognitive Walkthrough.....	132
Figure 4.15 : Onglet de description des expériences en CWE.....	132
Figure 4.16 : Onglet de description des scenarii.....	133
Figure 4.17 : Onglet de description des ressources.....	134
Figure 4.18 : Ecran d'analyse de tâche.....	135
Figure 4.19 : Ecran de choix de méthode.....	136
Figure 4.20 : Ecran d'inspection de tâche.....	137
Figure 4.21 : Problèmes par évaluateur.....	138
Figure 4.22 : Synthèse des problème par catégorie.....	138

Liste des figures

Figure 4.23 : Synthèse des problèmes par évaluateur.....	139
Figure 4.24 : Diagramme de paquetage.....	139
Figure 4.25 : Paquetage org.thomah.m2vc.core.....	140
Figure 4.26 : Paquetage org.thomah.cwe.view.....	140
Figure 4.27 : Diagramme de navigation écran.....	141
Figure 4.28 : Réalisation du cas d'utilisation « Définir les connaissances informatiques des utilisateurs ».....	143
Figure 4.29 : Réalisation du cas d'utilisation « Décrire le système à évaluer».....	144
Figure 4.30 : Réalisation du cas d'utilisation « Identifier les utilisateurs ».....	145
Figure 4.31 : Réalisation du cas d'utilisation « Définir les connaissances des utilisateurs concernant le système ».....	146
Figure 4.32 : Réalisation du cas d'utilisation : « Choisir les scénarii à évaluer ».....	146
Figure 4.33 : Réalisation du cas d'utilisation : « Analyser les tâches ».....	147
Figure 4.34 : Réalisation du cas d'utilisation : « Choisir la méthode d'évaluation ».....	148
Figure 4.35 : Réalisation du cas d'utilisation : « Inspecter les tâches ».....	149
Figure 4.36 : Réalisation du cas d'utilisation : « Calculer les tableaux de bord ».....	150
Figure 5.1 : Fiche problème (cf. aussi Annexe E).....	160
Figure 5.2 : Répartition des tâches selon les avis émis par chaque groupe.....	162
Figure 5.3 : Répartition des groupes d'évaluation selon les avis qu'ils ont émis.....	163
Figure 5.4 : Intersection des sources des problèmes détectés par les experts.....	163

## Liste des tableaux

Tableau 1.1 : Types de tâches dans CTT.....	24
Tableau 1.2 : Opérateurs de CTT.....	24
Tableau 2.1 : Synthèse de toutes les versions et variantes de CW.....	65
Tableau 2.2. Synthèse des études comparatives.....	71
Tableau 2.3 : Synthèse des études non comparatives.....	80
Tableau 2.4 : Citations et perspectives sur la spécificité des diverses versions et variantes.....	82
Tableau 3.1 Principes de collaboration.....	95
Tableau 3.2 : Les quatre catégories de lexiques.....	110
Tableau 3.3 : Tableau de couverture de scenarii.....	113
Tableau 3.4 Synthèse des problèmes d'utilisabilité par inspecteur.....	118
Tableau 3.5 Synthèse des problèmes d'utilisabilité.....	118
Tableau 3.6 Confrontation des résultats des inspecteurs.....	119
Tableau 4.1 : Les différents diagrammes utilisables dans chaque vue [Muller & Gaertner, 2002]	122
Tableau 4.2 : Correspondance des types d'attribut de classe et de types de colonne d'une entité....	124
Tableau 5.1 : Information et caractéristiques retenues.....	155
Tableau 5.2 : Les intervenants du premier groupe d'évaluateurs.....	157
Tableau 5.3 : Liste des exigences d'évaluation.....	158
Tableau 5.4 : Résultat des tests fonctionnels.....	162
Tableau 5.5 : Nombre de problèmes trouvés par chaque expert.....	163
Tableau 5.6 : Comparaison des problèmes issus de l'action numéro 4 et de la question numéro 4.	164
Tableau 5.7 : Comparaison des problèmes issus de l'action numéro 17 et de la question numéro 4 .....	165
Tableau 5.8 : Nombre de problèmes par catégorie et par expert.....	165

# Introduction générale

L'Interaction homme-machine est la discipline consacrée à la conception, la mise en œuvre et à l'évaluation des systèmes informatiques interactifs destinés à des utilisateurs, ainsi qu'à l'étude des principaux phénomènes qui les entourent [Hewett et al., 1996].

L'IHM est fondamentalement liée à des tâches de traitement de l'information. En interagissant avec le système interactif, l'utilisateur a des buts et des sous-butts spécifiques en tête. La capacité humaine pour prélever et traiter l'information n'est pas infinie : l'être humain a des limites que ce soit au niveau moteur, sensoriel, ou mental [Norman, 1986]. Pour que l'interaction soit efficace, la conception de l'interface homme-machine doit respecter les capacités humaines de traitement.

Une question fondamentale est donc la suivante : comment concevoir un système interactif qui respecte la capacité de traitement de l'utilisateur ?

En plus de connaissances sur les besoins relativement au système et en particulier aux fonctionnalités qu'il doit offrir, le concepteur a besoin de bien connaître l'utilisateur et son environnement ; ce qui doit donc conduire le concepteur à bien d'identifier les différentes capacités de l'utilisateur ainsi que ses limites.

La prise en compte des différentes caractéristiques de l'utilisateur nécessite des démarches méthodologique adaptées, résolument centrées utilisateur tout au long du projet. On trouve dans la littérature différents modèles de développement enrichis sous l'angle de l'interaction homme-machine [Kolski et al., 2001]. On constate que chaque modèle dit enrichi accorde beaucoup d'importance à l'évaluation. L'évaluation est le moyen de vérifier que le système qu'on conçoit satisfait les exigences et les caractéristiques de l'utilisateur. L'évaluation est donc un composant important dans le processus de conception des systèmes interactifs. De ce fait, elle a fait l'objet de nombreuses études et recherches depuis plus d'une trentaine années par la communauté en IHM. Il existe aujourd'hui de nombreuses méthodes et techniques d'évaluation ; plusieurs études ont été menées proposant des revues et synthèses de celles-ci.

L'inspection de l'utilisabilité est un nom générique attribué à une catégorie comprenant un ensemble de méthodes basées sur le fait qu'un ou plusieurs évaluateurs inspectent ou examinent les aspects de l'utilisabilité de l'interface utilisateur. Selon [Nielsen & Mack, 1994], l'utilisabilité est un concept assez large qui se rapporte à l'apprentissage et à l'utilisation du système par l'utilisateur, de même qu'à la satisfaction de l'utilisateur utilisant le système.

Cognitive Walkthrough (CW) fait partie des méthodes d'inspection de l'utilisabilité [Nielsen, 1994]. C'est une méthode qui associe le parcours (walkthrough) de l'interface à un modèle cognitif. L'évaluateur parcourt l'interface dans le contexte des tâches qu'un utilisateur typique aura besoin d'accomplir. Les actions et les réponses de l'interface homme-machine sont évaluées par rapport aux buts et connaissances de l'utilisateur, en répondant à des questions se rapportant au modèle cognitif de la méthode ; les écarts entre les attentes de l'utilisateur et les étapes requises par l'interface sont notés.

CW intéresse considérablement les chercheurs en évaluation ; l'existence de plusieurs versions et variantes en témoigne. Mais son utilisation est souvent limitée. Nous supposons que celle-ci est due à plusieurs causes. Par exemple, CW nécessite beaucoup de temps d'évaluation à cause de son caractère répétitif et la nécessité de répondre à des questions à chaque action d'une tâche. Donc pour un projet possédant une contrainte temporelle forte, l'équipe de développement peut être réticente à l'utiliser [Spencer, 2000]. En plus, CW nécessite des évaluateurs experts, ce qui n'est pas toujours possible dans certains projets, en raison par exemple de contraintes budgétaires ou de disponibilité.

Dans ce contexte, nous nous proposons dans le cadre de cette thèse de spécifier et réaliser un environnement d'aide à l'utilisation de la méthode Cognitive Walkthrough. Cet environnement vise à couvrir les différentes étapes de l'évaluation en partant de la phase de préparation jusqu'à celle d'analyse des résultats.

Dans le cadre de cette thèse, nous sommes amenés à passer en revue dans un premier temps différents aspects théoriques concernant les systèmes interactifs et leur évaluation. Ensuite, nous nous focalisons sur la méthode Cognitive Walkthrough. Ce qui nous permettra de spécifier l'environnement CWE (Cognitive Walkthrough Environment), d'en proposer un prototype et de procéder à une première évaluation de celui-ci.

Cette thèse est structurée en cinq chapitres.

Le premier chapitre recense les différents aspects théoriques concernant les systèmes interactifs. Après une section introductive stipulant que quatre concepts entrent en jeu lors de l'interaction entre un humain et une machine, ce chapitre aborde la conception et la modélisation de l'artefact, l'objet ou concept à manipuler pour l'exécution d'une tâche quelconque. Dans le contexte de cette thèse, la conception centrée utilisateur nous intéresse plus particulièrement. En ce qui concerne la modélisation de l'artefact, nous traitons celle-ci sous l'angle de la description de son identité et de celle de sa procédure d'utilisation.

La section 3 du chapitre 1 concerne les tâches, le deuxième concept impliquée dans l'IHM. Cinq modèles permettant de modéliser des tâches humaines sont décrites. Tout d'abord, « l'Analyse Hiérarchique de Tâche », la plus ancienne des méthodes, puis GTA (Groupware Task Analysis), une méthode permettant de modéliser et analyser les tâches coopératives. En troisième, on décrit une méthode analytique de description de tâche appelée MAD. Cette méthode permet en principe de décrire une tâche quel que soit son niveau de complexité. En quatrième, on décrit une autre méthode de modélisation hiérarchique des tâches appelée CTT (ConcurTaskTree). C'est une méthode puissante qui permet de modéliser à la fois les tâches simples et les tâches coopératives. Et enfin, on décrit la méthode MDA (Modèle de description de l'activité).

La section 4 du chapitre 1 concerne les utilisateurs, le troisième concept impliqué dans l'IHM. On y décrit quelques connaissances et modèles concernant les utilisateurs en commençant par la loi de Fitts et le modèle du processeur humain, un modèle qui représente l'utilisateur comme un système de traitement d'information disposant de mémoires et d'un processeur. Entre autre, le modèle CE+, un modèle de l'apprentissage par exploration, à la base de la méthode Cognitive Walkthrough ainsi que le concept de conscience de situation et de la représentation occurrente sont aussi décrits dans la section 4. Enfin, cette section se termine par la description d'un processus permettant de structurer les connaissances sur les utilisateurs appelé modélisation de l'utilisateur.

La section 5 est consacrée aux règles ergonomiques qui sont des ensembles de principes, des règles et conventions ou des recommandations. Tandis que la section 6 décrit quelques normes selon les quatre catégories : la qualité d'utilisation de l'artefact, la qualité de l'artefact, la qualité du processus de développement et la capacité de l'organisation.

Et enfin la section 7 décrit l'évaluation en tant qu'étape importante du processus de développement d'un système interactif. En effet l'évaluation permet de vérifier l'adéquation du système à l'utilisateur. Les critères de Nielsen ainsi que ceux proposés par Bastien et Scapin sont décrits. La méthode dite de tests d'utilisabilité en tant que méthode du type test utilisateur est décrite. Ensuite quelques généralités sur les méthodes du type inspection de l'utilisabilité dont la méthode Cognitive Walkthrough fait partie sont aussi décrites.

Le chapitre 2 est consacré à la méthode Cognitive walkthrough et un ensemble de versions et variantes de celle-ci. En effet à partir de la première version, Cognitive Walkthrough a évolué et a

donné plusieurs versions. Nous décrivons dans la section 2 les trois versions initiales. Par ailleurs nous avons trouvé aussi dans la littérature plusieurs variantes de la méthode. Ainsi nous décrivons dans la section 3 onze variantes représentatives : Heuristic Walkthrough (HW), Norman's Cognitive Walkthrough, Streamlined Cognitive Walkthrough, Cognitive Walkthrough for the Web, Groupware Walkthrough, Activity Walkthrough, Interaction Walkthrough, Cognitive Walkthrough with users, Extended Cognitive Walkthrough, Distributed Cognitive Walkthrough, Enhanced Cognitive Walkthrough. Dans la section 5, nous proposons quatre synthèses complémentaires de ces différentes versions et variantes. La première synthèse concerne les aspects conceptuels, méthodologiques et technologiques ; elle contribue à l'étude de la manière dont la méthode a évolué selon ces trois aspects. La deuxième synthèse concerne les études comparatives concernant les différentes versions et variantes de CW impliquant d'autres méthodes. La troisième synthèse concerne les études non comparatives sur les différentes versions et variantes de Cognitive Walkthrough. Elle récapitule les différentes études proposant des cas utilisant les différentes versions et variantes de la méthode. Et enfin la dernière synthèse propose à partir des trois premières synthèses une ébauche d'aide au choix d'une méthode parmi les différentes versions et variantes.

A partir des connaissances fournies par les deux chapitres précédents, notre contribution réside dans la spécification de l'environnement CWE (Cognitive Walkthrough Environment). C'est l'objet du chapitre trois. Il décrit dans un premier temps le processus d'évaluation selon notre besoin. Les différents acteurs de l'évaluation sont définis ainsi que les différentes étapes de l'évaluation. Ensuite le chapitre trois décrit les structures des différents entités constituant notre domaine. En plus de l'artefact, de l'utilisateur et de la tâche, il s'agit de modéliser la méthode d'évaluation et le résultat de l'évaluation. Et enfin, la section 4 du chapitre 3 décrit le comportement dynamique de l'environnement CWE. Ainsi les différentes étapes de l'évaluation en commençant par la phase de préparation jusqu'à l'analyse de résultats sont décrites dans ce chapitre.

Le chapitre 4 est consacré à la réalisation d'une maquette de l'environnement CWE selon la spécification décrite dans le chapitre 3. Ce chapitre commence par la description des objectifs architecturaux ainsi que les différentes contraintes que nous devons respecter. Nous décrivons entre autres, les exigences logicielles qui ont des impacts sur l'architecture de CWE, ainsi que les contraintes de développement.

Ensuite, nous décrivons l'architecture de l'environnement CWE en commençant par les décisions architecturales que nous avons prises conformément aux objectifs et contraintes décrits dans la section précédente. Puis nous décrivons l'architecture à travers les cas d'utilisation les plus significatifs. Nous présentons l'environnement CWE en termes de paquetages logiciels. Nous présentons aussi quelques quelques pages écrans de l'IHM de CWE en partant des cas d'utilisation les plus significatifs, ainsi que la réalisation de tels cas.

Dans le chapitre 4, nous décrivons aussi le mécanisme de persistance des données que nous utilisons conformément aux objectifs et contraintes architecturaux précédemment décrits.

Cette thèse concerne le domaine de l'évaluation, il est donc légitime de pratiquer l'évaluation sur la maquette réalisée dans le cadre de cette thèse. Le chapitre 5 est ainsi consacré à l'évaluation de la maquette de l'environnement CWE décrite dans le chapitre 4. Dans un premier temps, nous décrivons le processus d'évaluation que nous adoptons. Ensuite nous décrivons et analysons les résultats des évaluations effectuées par des sujets novices en évaluation d'IHM (étudiants en master informatique). Puis nous décrivons et analysons les résultats des évaluations effectuées par trois experts en évaluation utilisant la méthode Cognitive Walkthrough.

Pour terminer cette thèse, nous donnons quelques conclusions et perspectives en vue d'améliorer l'environnement CWE d'une part et d'orienter les recherches futures d'autre part.

# Chapitre 1 - Eléments clés liés à la prise en compte des facteurs humains dans la conception et l'évaluation d'IHM

---

## Plan du chapitre

1.1- Introduction.....	2
1.2- Les artefacts.....	3
1.2.1 - La conception centrée utilisateur.....	3
1.2.2 - Modèle de l'artefact.....	4
1.3- Les tâches et leur modélisation.....	5
1.3.1 - Hierarchical Task Analysis.....	6
1.3.2 - Groupware Task Analysis(GTA).....	7
1.3.3 - Méthode Analytique de Description de tâches (MAD).....	8
1.3.4 - CTT.....	9
1.3.5 - Modèle de description de l'activité (MDA).....	11
1.4- Les utilisateurs et leur modélisation.....	12
1.4.1 - La loi de Fitts.....	13
1.4.2 - Le modèle du processeur humain.....	13
1.4.3 - GOMS.....	14
1.4.4 - CCT.....	14
1.4.5 - Le modèle de Rasmussen.....	15
1.4.6 - Le modèle de Norman.....	15
1.4.7 - Le modèle de Kintsch.....	16
1.4.8 - Le modèle EXPL.....	16
1.4.9 - Le modèle CE+.....	17
1.4.10 - Le modèle LICAI.....	18
1.4.11 - Le modèle CoLiDeS.....	18
1.4.12 - La conscience de la situation et la représentation occurrente.....	19
1.4.13 - Modélisation de l'utilisateur.....	19
1.5- Règles ergonomiques.....	20
1.5.1 - Généralités.....	20
1.5.2 - Normes liées à l'IHM.....	21
1.6- Evaluation des systèmes interactifs : principes et critères.....	26
1.6.1 - Les heuristiques de Nielsen.....	27
1.6.2 - Les critères ergonomiques de Bastien et Scapin.....	28
1.6.3 - Méthodes d'évaluation des systèmes interactifs.....	31
1.7- Conclusion.....	33

---



## 1.1 Introduction

Comme rappelé dans l'introduction générale, selon [Hewett et al., 1996], l'Interaction Homme-Machine est une discipline qui s'occupe de la conception, l'évaluation et la mise en œuvre de systèmes informatiques interactifs à usage humain et à l'étude des principaux phénomènes qui les entourent. C'est donc une discipline qui concerne la plupart de nos activités quotidiennes. En effet, l'informatique est sortie du milieu professionnel en s'intégrant dans le milieu domestique. Ceci est dû d'une part au développement spectaculaire d'Internet et d'autre part au développement des technologies émergentes telles que la communication mobile. Ainsi les informations peuvent être disponibles partout et à partir des différents dispositifs plus personnels comme les téléphones portables, les appareils de guidage GPS, etc.

Ainsi, il ne suffit plus de concevoir et de réaliser un système performant mais un système qui facilite les interactions avec l'utilisateur humain, c'est-à-dire un système qui prend en compte les facteurs humains.

La prise en compte des facteurs humains vise à assurer un développement équilibré des aspects techniques et humains liés aux machines et systèmes interactifs. Elle fournit un processus qui assure l'application de la connaissance scientifique au sujet des caractéristiques humaines par les spécifications, la conception et l'évaluation du système [Mod, 2000].

Quel que soit le processus adopté, la conception d'un système interactif est un processus itératif mettant en jeu quatre entités (voir figure 1.1) à savoir le système (que nous dénommons désormais artefact), l'utilisateur, la tâche et l'environnement. L'artefact est une entité physique ou conceptuelle conçue pour une catégorie spécifique d'utilisateurs qui doivent accomplir une tâche donnée [Boy, 2003].

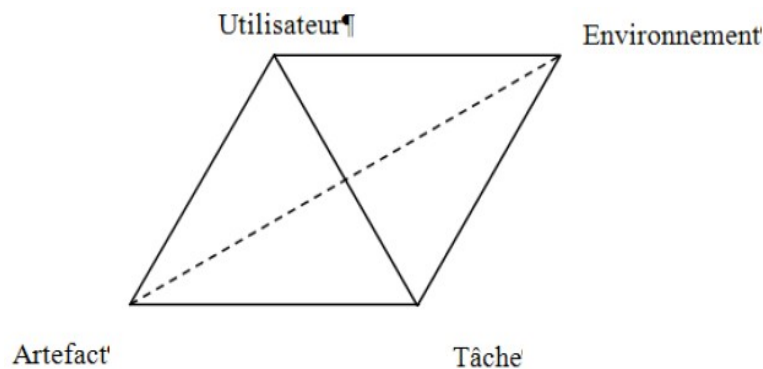


Figure 1.1 : Pyramide artefact - utilisateur - tâche - environnement

La prise en compte des facteurs humains conduit à deux types d'approches [Dejours, 1995] :  
(a) la recherche des origines et moyens de contrôler la défaillance humaine en situation de travail ;  
(b) la recherche de modèles et de moyens pour modéliser, développer et gérer les ressources (humaines et machines) dans le travail.

Certains chercheurs adoptent la première approche qui fait l'hypothèse que l'on peut tout caractériser et ont proposé des règles et des propositions de bon sens établies expérimentalement pour aider les développeurs à prendre en compte les facteurs humains lors du développement et de l'évaluation des systèmes interactifs. Ces règles sont généralement appelées règles ergonomiques [Stephanidis & Akoumianakis, 1999].

La seconde approche considère que les situations ne sont pas descriptibles dans leur intégralité, et que leur modélisation est nécessaire. Les sciences cognitives proposent à ce sujet un ensemble de théories et modèles sur l'homme en situation de travail.

Dans ce chapitre, nous allons voir les quatre entités selon les points de vue suivants :

- Le point de vue théorique, selon lequel sont parcourues plusieurs théories pertinentes sur l'entité,
- Le point de vue normatif et réglementaire ; celui-ci est associé à des normes et règles ergonomiques concernant l'entité,
- Le point de vue modélisation pour l'évaluation. Ici dans le cadre de l'évaluation, on fait le point sur les informations pertinentes pour la conduite de l'évaluation.
- Ensuite, il s'agit de faire le point sur l'évaluation de système interactif.

## **1.2 Les artefacts**

Le système est l'artefact qu'on manipule pour exécuter une tâche. Pour écrire cette thèse par exemple, on a utilisé un système de traitement de texte. Pour commander par internet un ou plusieurs articles dans le catalogue des 3 Belges, on utilise l'application qui se trouve à l'adresse <http://www.3belges.fr>.

Il y a plusieurs moyens de réaliser un artefact. Mais en vue de réaliser un artefact utile et utilisable, c'est-à-dire qui prend en compte les facteurs humains, la conception et la réalisation d'un artefact doivent suivre une démarche globale centrée utilisateur.

Nous allons nous intéresser dans un premier temps aux principes de la conception centrée utilisateur. Puis on va voir comment modéliser les artefacts, et enfin on citera quelques règles ou guides concernant les principes de la conception centrées utilisateur et la modélisation de l'artefact.

### **1.2.1 La conception centrée utilisateur**

La conception centrée utilisateur consiste à considérer les utilisateurs et leurs besoins tout au long du processus de développement d'un artefact [Norman, 1986]. Cette approche de conception repose sur trois principes centraux : une attention portée sur les utilisateurs et leurs tâches tout au long du processus de conception, la mise en œuvre de mesures empiriques, et une conception itérative [Gould & Lewis, 1985 ; Gould, 1988].

Le premier principe suggère qu'il faut comprendre les utilisateurs. On ne se contente pas de les identifier, ni de les décrire, ni de les stéréotyper. Ce qui nécessite des études des caractéristiques cognitives, émotionnelles, comportementales, anthropométriques et des attitudes des utilisateurs, ainsi que des études sur la nature des tâches prévues. Ces études sont souvent réalisées à travers des interviews et des observations auprès de panels d'utilisateurs membres de l'équipe de projet, qui soient dans la mesure du possible représentatifs des utilisateurs finaux.

Le deuxième principe quant à lui suggère que l'exécution des travaux réels devrait être observée, analysée et enregistrée, y compris en utilisant des simulations et des prototypes. Ceci nécessite des mesures comportementales réelles de l'utilisabilité et de la facilité d'utilisation, en attribuant aux utilisateurs des tâches d'essai simples permettant d'enregistrer et d'analyser l'exécution, les avis et les attitudes des utilisateurs; on parle à ce sujet de tests utilisateurs.

Enfin, on peut dire que le troisième principe est une conséquence du deuxième. En effet, les mesures et observations suggérées par le deuxième principe peuvent évoquer des problèmes qu'il faut corriger.

A partir de ces principes découlent plusieurs méthodes de conception, Vredenburg et al. [2002] ont par exemple effectué une étude de synthèse des différentes pratiques en matière de conception centrée utilisateur.

Mais selon Gould et Lewis [1988], l'approche doit contenir une première phase de conception qui doit précéder n'importe quelle itération de conception. Son but est de produire une spécification préliminaire de l'artefact, de rassembler des profils utilisateur, et de spécifier les tâches, les contextes d'utilisation, et de procéder à des mesures d'intérêt. Ensuite la phase itérative commence ; elle comprend en particulier l'examen des buts comportementaux, l'évaluation continue, la modification de l'interface homme-machine, le prototypage rapide et flexible et la réalisation fortement modulaire.

## **1.2.2 Modèle de l'artefact**

Un artefact est un système complexe résultat d'un processus de conception complexe. Boy [2003] propose un moyen de modéliser un artefact. Il s'agit de décrire l'identité de l'artefact, puis de documenter l'historique de la conception de l'artefact, ensuite de décrire les procédures d'utilisation de l'artefact et enfin de documenter le processus d'évaluation de l'artefact.

Parmi ces quatre rubriques du modèle de l'artefact, nous pensons que la première et la troisième sont suffisantes pour que les utilisateurs finaux puissent interagir avec l'artefact. Nous allons donc décrire les concepts les plus importants de ces deux rubriques.

Concernant la rubrique identité, le domaine, la motivation et le contexte d'utilisation de l'application ou artefact sont très importants.

Le domaine de l'application est le domaine ou le métier dans lequel l'application sera exploitée ou utilisée. Par exemple, le domaine du traitement de texte utilisé pour la rédaction de cette thèse est la bureautique ; le domaine de l'application 3Belges citée plus haut est le commerce électronique.

La compréhension du domaine de l'application est primordiale lors de l'élaboration de l'application. C'est pourquoi le cycle de vie d'une application comprend toujours une phase d'analyse des besoins qui consiste à comprendre et modéliser le domaine de l'application. Prenons l'exemple du Processus Unifié [Jacobson et al., 1999] qui est une méthode générique de développement de logiciel orienté objet, itérative et incrémentale. Nous avons l'« expression des besoins » parmi les itérations correspondant aux différentes étapes de l'enchaînement d'activités et la « création » ou « analyse des besoins » parmi les incréments correspondant aux avancées dans les différents stades de développement.

La compréhension du domaine de l'application dépend de la compréhension des différents termes et concepts de l'application. Donc elle influence l'utilisation de l'application. Par exemple, concernant l'application interactive « comment ça marche » (Editions Nathan), des experts l'ayant évalué (évaluation décrite dans [Huart, Kolski et Sagar, 2004]) ont trouvé un problème d'utilisabilité lié à la mauvaise compréhension du terme machine ; selon eux les enfants de 8 à 14 ans, utilisateurs cibles de l'application qui souhaitent savoir comment fonctionne un ordinateur ne doivent pas nécessairement connaître qu'un ordinateur est une machine.

Pour pouvoir effectuer convenablement une évaluation d'IHM, l'évaluateur doit connaître aussi les différents termes et concepts de l'application.

Une application est exploitée en vue de réaliser des tâches précises dans un contexte bien défini. Chaque contexte amène les concepteurs à utiliser des techniques particulières pour prendre en compte certains facteurs spécifiques. Pour une application pour un terminal de point de vente, par exemple, qui est exploitée dans un contexte commercial, la vitesse d'exécution est un facteur dont il faut tenir compte afin d'éviter la queue aux caisses due aux temps de traitement d'une transaction. Ces contextes peuvent être classifiés de plusieurs manières selon différents points de vue. Chaque classe de contexte correspond à une motivation.

Dans [Glass et Versey, 1995], on trouve par exemple quelques classifications des applications informatiques : des classifications selon le point de vue des industriels (IBM, Digital), des classifications selon le point de vue des communautés scientifiques (ACM, IEEE) et des classifications selon le point de vue académique.

Dans [Shneiderman, 1998], on trouve 4 motivations :

- **La motivation critique** correspond à celle des applications de contrôle du trafic aérien, de surveillance d'une réaction nucléaire, de monitoring dans un service de soins intensifs, etc. Pour les applications à motivation critique, la sécurité et l'efficacité sont des facteurs primordiaux au détriment des coûts et des satisfactions subjectives.
- **La motivation industrielle et commerciale** correspond à celle des applications des banques et assurances, des réservations d'hôtel et de vol. Pour ces applications, les coûts et les performances sont des facteurs dont il faut tenir compte. Ainsi, elle exige un temps d'apprentissage limité, une grande rapidité d'exécution et une couverture adéquate.
- **La motivation domestique et de loisir** correspond à celle des applications de bureautique, des diverses applications domestiques sur des ordinateurs personnels, des jeux, etc. La satisfaction individuelle et la rapidité d'apprentissage sont des facteurs à privilégier.
- **La motivation créative et coopérative** correspond à celle des systèmes d'information et d'aide à la décision, des différents types d'applications dont la réalisation est assistée par ordinateur ou d'activités à caractère artistique, des applications exploratoires comme l'encyclopédie électronique, la recherche d'information sur internet et les applications coopératives permettant à plusieurs personnes de travailler ensemble même si elles sont physiquement séparées dans le temps et dans l'espace. Dans ce cas, la satisfaction individuelle devient le critère majeur à vérifier.

En ce qui concerne l'évaluation, la motivation de l'application peut aider l'évaluateur à choisir la méthode d'évaluation à utiliser. Ainsi, par exemple, pour une application de motivation domestique ou de loisir qu'on utilise souvent sans formation et qu'on apprend en l'utilisant, la méthode d'inspection Cognitive Walkthrough [Lewis et al, 1990], détaillée dans le chapitre 2, sera adéquate.

### 1.3 Les tâches et leur modélisation

Une tâche est une activité exécutée par une personne ou un groupe de personnes qui coopèrent, en vue d'atteindre un but. Le but correspond à un changement d'état désiré dans un domaine donné. Par exemple, dans le but de posséder un pantalon Daddydas gris au meilleur prix, un homme adulte peut réaliser la tâche suivante : « Acheter un pantalon Daddydas gris pour homme adulte au meilleur prix ».

Une tâche peut être abstraite ; c'est la représentation mentale de l'activité que la personne a pour atteindre son but. L'exemple que nous venons de citer est par exemple une tâche abstraite. Pour réaliser cette tâche, la personne peut utiliser l'application qui se trouve à l'adresse <http://www.3belges.fr> ou celle qui se trouve à l'adresse <http://www.padedoute.fr>. En d'autres termes, la personne doit passer à l'acte en interagissant avec son environnement par l'intermédiaire d'une application en vue de provoquer ce changement d'état qui correspond à son but.

La tâche supportée par une application pour atteindre le but d'une personne est appelée tâche implémentée. Pour notre exemple, la tâche implémentée dans l'application 3belges est la tâche composée des actions suivantes :

- Cliquer sur Sport Forme,
- Cliquer sur Textile Homme,
- Cliquer sur Pantalons, Pantacourts, Shorts,
- Choisir la marque Daddydas,
- Choisir un prix,
- Choisir la couleur gris, la taille et la quantité Un,
- Cliquer sur Ajouter au panier
- Payer

Pour passer d'une tâche abstraite à une tâche implémentée, les concepteurs ou analystes doivent faire de l'analyse de tâches. Et pour faciliter l'analyse, ils utilisent un cadre ou modèle appelé modèle de tâche. Ce dernier permet de formaliser les buts des utilisateurs, la manière dont ils seront atteints et les ressources et informations nécessaires pour les atteindre.

Le modèle de tâches peut être utilisé de plusieurs manières dans le cycle de vie d'une application. On peut l'utiliser pendant la spécification et la conception puisqu'il permet de décrire les tâches ainsi que les ressources utilisées. C'est le cas de TKS [Johnson, 1992] et de CTT [Paternò, 2000]. On peut aussi l'utiliser pendant l'évaluation : GOMS [Card, Moran & Newell, 1983] permet par exemple d'évaluer la performance humaine.

Il existe aujourd'hui plusieurs modèles de tâches. La plupart sont décrits dans [Diaper et Stanton, 2004]. Nous allons en décrire plusieurs représentatifs selon les concepts qu'ils utilisent, leurs méthodes de planification, le plus bas niveau de décomposition des tâches, le niveau de décomposition où les actions ont lieu, le support des tâches collaboratives et la disponibilité des outils.

### 1.3.1 Hierarchical Task Analysis

HTA est le plus ancien des modèles de tâche. Il a été décrit pour la première fois dans [Annett et Duncan, 1967]. Il utilise juste les concepts de tâches et sous-tâches, de buts et sous-butts et de plans. A chaque tâche correspond un but qui sera atteint si la tâche est exécutée. Il permet donc de décrire une tâche quelconque sous forme de hiérarchie de tâches et sous-tâches, et de plans. Les plans décrivent les façons d'exécuter les tâches pour atteindre les buts. Les feuilles de l'arbre sont donc des tâches où les actions ont lieu. D'après [Annett et al., 2000], HTA peut supporter l'analyse des tâches collaboratives. Selon [Stanton, 2006], il n'existe pas d'outils qui couvrent l'ensemble du processus de HTA. L'outil proposé par [Bass, 1995] simplifie seulement la production des diagrammes en permettant la manipulation directe des objets des diagrammes. D'autres outils

comme *OrgPLUS* et *Inspiration* ont été développés, mais ils ne couvrent tous les aspects de HTA. La figure 1.2 montre notre tâche en utilisant le modèle HTA.

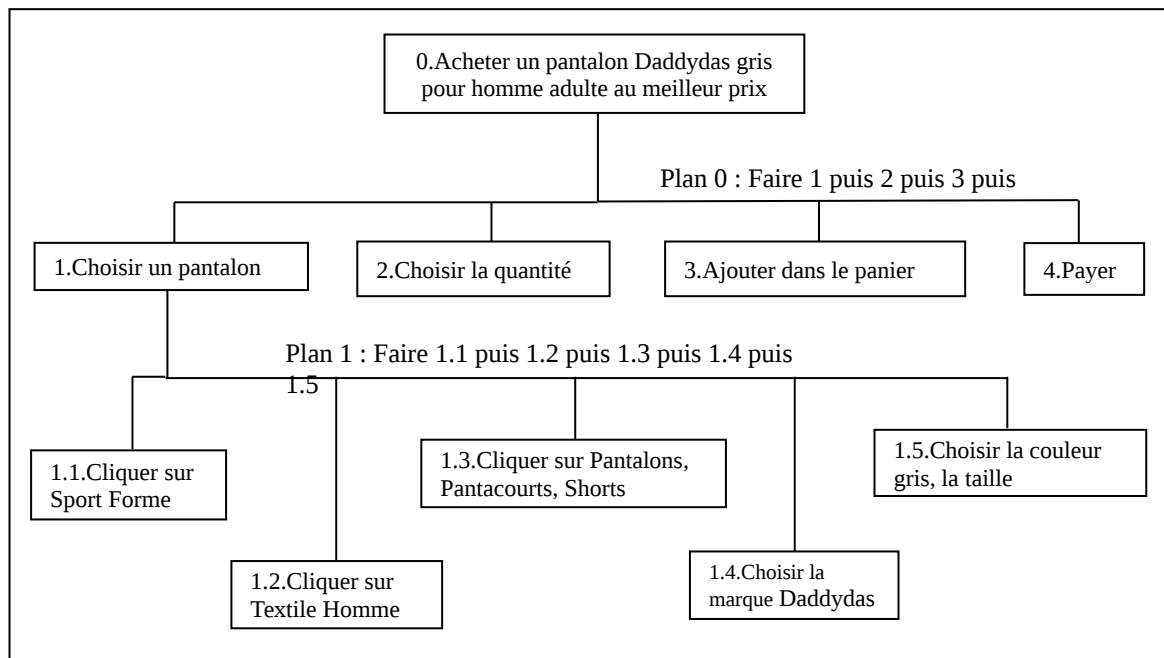


Figure 1.2 : Exemple de diagramme HTA

### 1.3.2 Groupware Task Analysis (GTA)

Le modèle de tâche GTA [Van der Veer, 1996 ; Van der Veer, 1999] permet de produire trois catégories de modèles selon la situation dans le processus de création de l'IHM. Le modèle 1 décrit les tâches actuelles. Le modèle 2 décrit les tâches du système futur. En d'autres termes, il représente le système futur du point de vue de la structure des tâches et de l'organisation du travail. Sa construction tient compte des conflits créés par le modèle 1 et est en accord avec les souhaits du client. Le troisième modèle décrit la machine virtuelle utilisateur. Il modélise le système futur en termes de technologie.

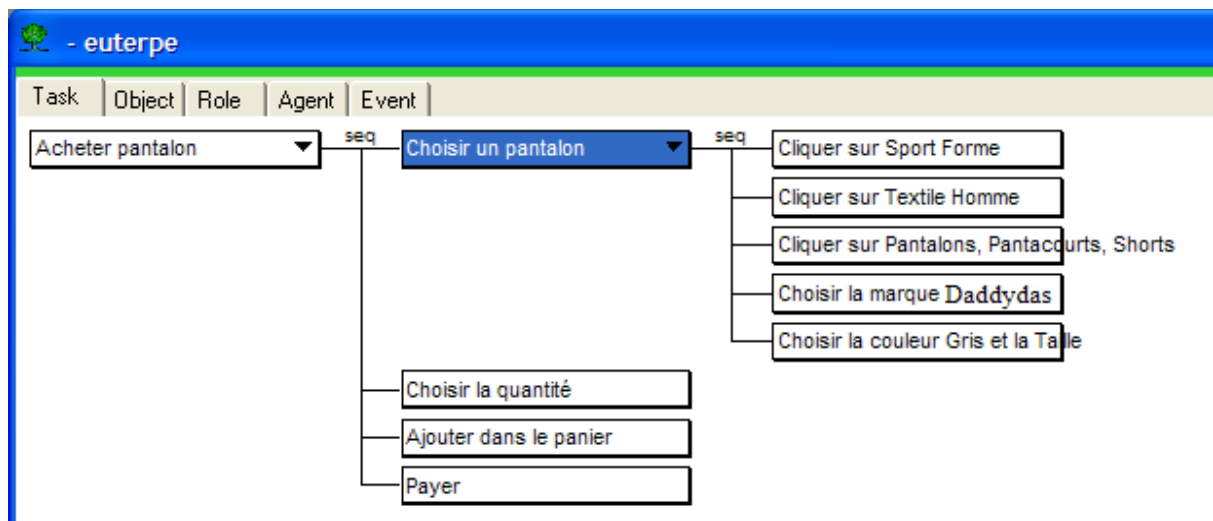


Figure 1.3 : Exemple de diagramme avec Groupware Task Analysis

Pour le modèle 1, GTA fournit un cadre de conception permettant de prendre en compte les trois aspects fondamentaux : les utilisateurs, leurs travaux et l'environnement de travail. La figure 1.3 présente le diagramme hiérarchique des tâches pour la tâche : « Acheter un pantalon Daddydas gris pour homme adulte au meilleur prix ». Ainsi, le modèle GTA repose sur les concepts suivants : objet, agent, rôle, événement et tâche.

Les objets sont des entités physiques ou non que les individus utilisent quand ils exécutent des travaux. Les agents sont des entités actives qui réalisent des tâches. Chaque agent a son propre rôle dans la réalisation d'une tâche donnée. Un agent donc peut être un seul individu, un groupe d'individu ou un programme informatique. Par contre le concept agent ne désigne pas un individu spécifique mais un ensemble d'individus possédant une certaine caractéristique commune. Un rôle correspond à un ensemble significatif de tâches exécutées par un ou plusieurs agents. Une tâche est significative si elle a un but clair ou si elle se distingue entre plusieurs groupes d'agents. Un événement est un changement d'état qui se produit à un moment donné lors de l'exécution d'une tâche. Le changement peut concerner les autres concepts (objet, agent, rôle et tâche) et peut entraîner l'exécution d'autres tâches. Une tâche est une activité réalisée par des agents pour atteindre un but. Une tâche complexe peut être décomposée en plusieurs sous-tâches. L'ordonnancement des tâches est décrit par un constructeur. Une tâche peut être aussi de base ou unitaire. Une tâche unitaire est une tâche de niveau le plus bas que l'individu considère en se référant à son travail. Une tâche basique est une tâche de plus bas niveau qui est définie par l'outil utilisé pour l'exécution du travail. Une tâche unitaire devrait être accomplie uniquement en exécutant une ou plusieurs tâches basiques. Une tâche basique ou une tâche unitaire peut être décomposée en action (utilisateur) ou opération (système). Une action est un composant identifiable d'une tâche basique ou d'une tâche unitaire qui a une signification en exécutant une unité de travail mais qui tire sa signification uniquement à partir de la tâche dont elle fait partie.

En résumé, pour le modèle GTA, le moyen de planification est le constructeur, le plus bas niveau de décomposition comprend les tâches basiques ; le niveau de décomposition concernant les actions intègre les actions utilisateurs et les opérations système. Le modèle GTA permet de représenter des tâches coopératives. Il dispose d'un outil appelé *Euterpe* [Van Weillie, 1998].

La figure 1.3 montre notre tâche en utilisant le modèle GTA.

### 1.3.3 Méthode Analytique de Description de tâches (MAD)

MAD\* [Scapin et Bastien, 2001] est un modèle de tâche basé sur un formalisme de décomposition hiérarchique de tâches. MAD\* est issue de MAD [Pierret-Golbreich, 1989]. Il est basé sur les concepts de tâche, constructeurs (ou opérateurs), utilisateur et objet.

Selon le formalisme MAD, une tâche permet de représenter un traitement quelconque quelque soit son niveau de complexité. Une tâche est caractérisée par trois groupes d'éléments : les éléments d'identification, les éléments de structure et les éléments d'attributs.

Les éléments d'identification sont : le numéro de la tâche et le nom de la tâche ;

Les éléments de structure sont :

- L'état-initial : sous-ensemble de l'état du monde constitué de la liste des objets, arguments d'entrée de la tâche ;
- L'état-final : sous-ensemble de l'état du monde constitué de la liste des objets, arguments de sortie de la tâche. Il s'agit des objets directement créés ou modifiés

suite à l'exécution de la tâche. Certains objets peuvent naturellement apparaître à la fois en entrée et en sortie ;

- Le but : sous-ensemble de l'état-final, indiquant explicitement le but recherché dans l'exécution de la tâche ;
- Les préconditions : ensemble des prédicats exprimant des contraintes sur l'état initial et qui doivent nécessairement être satisfaites pour pouvoir déclencher l'exécution de la tâche.
- Les postconditions : ensemble des prédicats exprimant des contraintes sur l'état final et qui doivent nécessairement être satisfaites après l'exécution de la tâche.
- Le corps de la tâche : une structure opérationnelle indiquant la méthode d'exécution de la tâche

Les éléments d'attributs sont :

- FAC : tâche facultative ;
- @ : tâche en boucle ;
- PRIOR : tâche prioritaire ;
- INTER : tâche interruptible ;
- La priorité de la tâche

Il existe deux types de tâches selon la nature du corps : la tâche simple et la tâche composée. Une tâche simple est une tâche dont le corps est constitué d'une entité procédurale insécable appelée action. En revanche, une tâche composée est une tâche composée de plusieurs tâches. Son corps est donc constitué d'une structure caractérisée par la liste des tâches qui la composent et d'un constructeur qui décrit l'agencement des tâches impliquées.

On distingue quatre catégories de constructeurs : les constructeurs de synchronisation (SEQ comme Séquentiel, PAR comme parallèle, SIM comme Simultané), les constructeurs d'ordonnancement (ET, OU, Alternatif), les constructeurs temporels (début, fin, durée) et les constructeurs auxiliaires.

MAD\* est supporté par le logiciel ALACIE (Atelier Logiciel d'Aide à la Conception Ergonomique) [Gamboa & Scapin, 1997].

### 1.3.4 CTT

CTT (ConcurTaskTree) [Paternò, 2000] est aussi un formalisme de décomposition hiérarchique des tâches, basé sur les concepts de tâches, objets, actions, constructeurs et rôles. D'après ce modèle, il existe quatre catégories de tâches :

- Les **tâches de l'utilisateur** : ce sont des tâches exécutées entièrement par les utilisateurs. Elles requièrent donc des activités cognitives ou physiques sans interagir avec le système,
- Les **tâches de l'application** : ce sont des tâches où seul le système intervient. On peut citer comme exemples la compilation d'un programme ou la réception de message via le réseau,
- Les **tâches d'interaction** : ce sont des tâches exécutées par les utilisateurs en interagissant avec le système. Les interactions sont déclenchées par les utilisateurs eux-mêmes. Lancer une requête sur une base de données en est un exemple,



- Les **tâches abstraites** : ce sont des tâches de plus haut niveau qui ne font pas partie des trois catégories précédentes. Elles se décomposent en sous-tâches pouvant appartenir à l'une des 4 catégories.

Tableau 1.1 : Types de tâches dans CTT






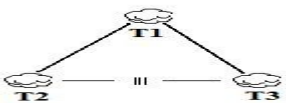
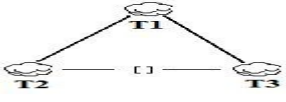
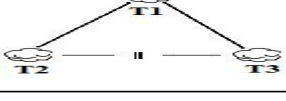
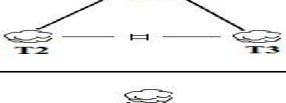


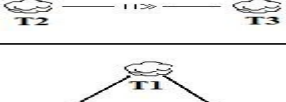
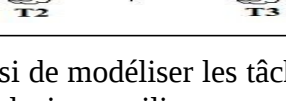
Symbole	Signification
	Tâche abstraite
	Tâche de l'application
	Tâche de l'utilisateur
	Tâche d'interaction
	Tâche coopérative

Tableau 1.2 : Opérateurs de CTT

Plan	Représentation graphique	Signification
<b>Concurrence</b> (T1   T2)		Les actions appartenant à ces deux tâches peuvent être exécutées dans n'importe quel ordre sans aucune contrainte spécifique.
<b>Choix</b> (T1 ]T2)		On peut choisir entre les deux tâches celle qu'il faut exécuter ; celle qui n'est pas choisie reste inactive jusqu'à ce qu'au moins l'exécution de l'autre soit terminée.
<b>Concurrence avec synchronisation</b> (T1[ ]T2)		Les deux tâches peuvent être exécutées simultanément tout en se synchronisant. Ce sont des tâches collaboratives qui doivent coopérer ou se synchroniser.
<b>Indépendance d'ordre</b> (T1 =T2)		Les deux tâches peuvent être exécutées dans n'importe quel ordre. Mais les deux ne s'exécutent pas simultanément.
<b>Désactivation</b> (T1]>T2)		La première tâche T1 est définitivement désactivée dès que la première action de la deuxième tâche T2 est exécutée.
<b>Activation</b> (T1]>>T2)		La deuxième tâche est exécutée dès que l'exécution de la première tâche est terminée.
<b>Activation avec passage d'information</b> (T1 ]>>T2)		La première tâche T1 fournit de l'information à la deuxième tâche T2 tout en l'activant.
<b>Suspension Reprise</b> (T1]>T2)		La deuxième tâche T2 a la possibilité d'interrompre la première tâche T1 et dès que l'exécution de T2 est terminée, l'exécution de T1 continue.

CTT permet aussi de modéliser les tâches coopératives, c'est à dire les tâches qui impliquent des actions de deux ou plusieurs utilisateurs pour être exécutées.

A chaque catégorie correspond une icône graphique comme le montre le tableau 1.1.

Par opposition aux trois précédents modèles, les constructeurs agissent au niveau des tâches sœurs. Il faut noter aussi que le modèle CTT possède plusieurs constructeurs comme le montre le tableau 1.2.

Ainsi, notre exemple se présente avec CTT selon la figure 1.4.

En outre, CTT fournit les moyens pour décrire les tâches coopératives. Ainsi, pour décrire une telle tâche, le modèle dispose de deux types d'arbres de tâches, l'un pour la partie coopérative et un autre pour chaque rôle qui est impliqué dans la tâche.

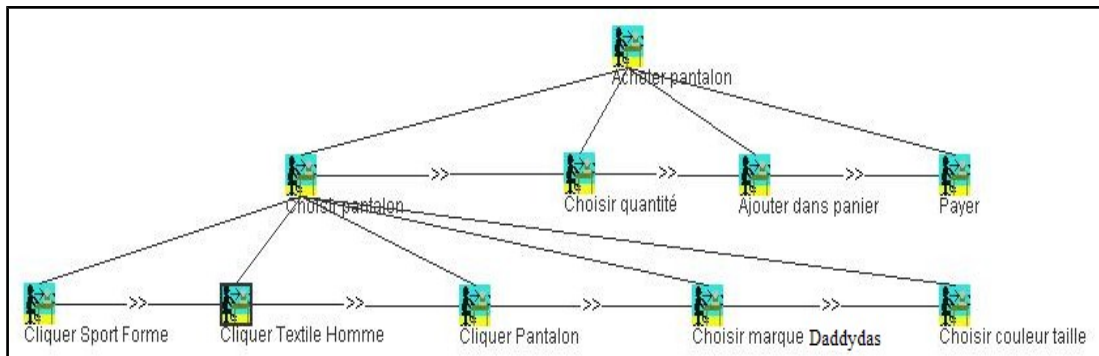


Figure 1.4 : Exemple de diagramme avec CTTE

### 1.3.5 Modèle de description de l'activité (N-MDA)

N-MDA (Noyau du Modèle de Description de l'Activité) [Lucquiaud, 2005] est un modèle issu des concepts de certains modèles que nous avons vu plus haut, à savoir : les tâches, les utilisateurs, les événements, et les objets de l'utilisateur. C'est un modèle orienté utilisateur qui est prévu pour servir de point d'entrée à d'autres modèles. De ce fait, il est composé seulement de deux modules : l'un permet de manipuler les tâches de l'utilisateur et l'autre s'occupe de la gestion des objets qui constituent l'environnement de cet utilisateur.

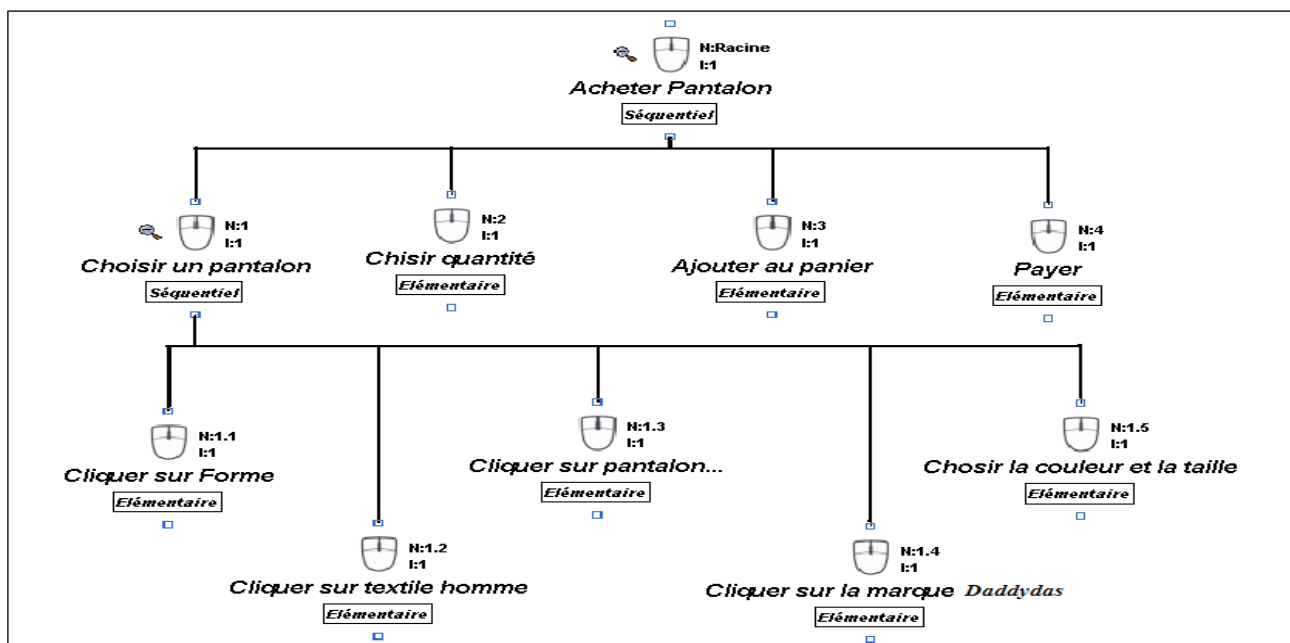


Figure 1.5 : Exemple de diagramme avec K-MAD

Ainsi par exemple, les données à récolter concernant les tâches sont réparties en quatre catégories : les informations sur la tâche (nom, but, durée) ; les caractéristiques de la tâche (fréquence et importance de la tâche) ; l'ordonnement local de la tâche (itération et interruptibilité de la tâche) et enfin l'ordonnement global de la tâche (opérateurs ou constructeurs SEQ et PAR).

N-MDA dispose d'un outil appelé K-MADe (Kernel of Model for Activity Description Environment) [Baron et al., 2006]. Notre tâche « Acheter un pantalon... » se présente selon la figure 1.5 avec cet outil. Notons que l'outil<sup>1</sup> ne prend pas en charge les tâches collaboratives.

## 1.4 Les utilisateurs et leur modélisation

En interagissant avec le système, l'utilisateur a des buts et des sous-buts spécifiques dans son esprit. Mais la capacité humaine pour prélever et traiter l'information n'est pas infinie : l'humain a des limites que ce soit au niveau moteur, sensoriel, ou mental.

Les connaissances sur les utilisateurs sont issues de plusieurs disciplines scientifiques, ce qui rend le domaine de l'IHM nécessairement multidisciplinaire. Ces connaissances couvrent les caractéristiques de l'homme nécessaires pour une conception si possible optimale de l'IHM ; c'est-à-dire une IHM qui produit une performance et une satisfaction maximum lors de l'exécution des tâches interactives. Ces connaissances concernent donc les aspects perceptuels, cognitifs, émotionnels, moteurs, et sociaux de l'informatique.

Nous ne pouvons pas prétendre décrire ici toute les connaissances (plusieurs prenant la forme de modèle) concernant les utilisateurs ; nous allons juste en décrire quelques unes ayant marqué l'évolution de l'IHM, par, exemple des architectures cognitives<sup>2</sup>.

### 1.4.1 La loi de Fitts

La loi de Fitts [Fitts, 1954] est un modèle du mouvement humain, prédisant le temps requis pour aller rapidement d'une position de départ à une zone finale de destination, en fonction de la distance à la cible et de la taille de celle-ci. La loi de Fitts est utilisée pour modéliser l'acte de "pointer", à la fois dans le vrai monde, par exemple avec une main ou un doigt, et en interagissant avec un ordinateur, par exemple avec une souris.

### 1.4.2 Le modèle du processeur humain

Le modèle du processeur humain (MHP=Model of Human Processor) a été proposé par Card, Moran et Newell [1983] et constitue le fondement des travaux sur la notion d'architecture cognitive. Selon ces auteurs, l'individu peut être représenté comme un système de traitement d'information régi par des règles assemblées à partir de résultats provenant de recherches expérimentales en psychologie. Ainsi le processeur humain comprend trois sous-systèmes interdépendants :

- le système perceptif comprend un ensemble de sous-systèmes spécialisés chacun dans le traitement d'une classe de stimuli.

---

<sup>1</sup>Il peut être vu comme une évolution de MAD et MAD\*, D.L. Scapin étant, comme pour ces deux modèles, un des promoteurs de N-MDA

<sup>2</sup>Selon Ritter et Young [2001], l'architecture cognitive est l'incarnation de l'hypothèse scientifique sur l'aspect de la cognition humaine relativement constant à travers le temps et relativement indépendant de la tâche. L'architecture est conçue pour simuler l'intelligence humaine.

- le système moteur s'occupe de suites de micro mouvements correspondant à la manipulation des dispositifs d'entrée.
- le système cognitif comprend une mémoire à long terme et une mémoire à court terme.
- Chacun de ces sous-systèmes dispose d'une mémoire et d'un processeur dont les performances se caractérisent à l'aide de paramètres. Pour une mémoire, les paramètres essentiels sont :
- $\mu$ , la capacité (c'est-à-dire le nombre d'éléments d'information mémorisés),
- $\delta$ , la persistance (c'est-à-dire le temps au bout duquel la probabilité de retrouver un élément d'information est inférieure à 0.5),
- $\kappa$ , le type d'information mémorisée (physique, symbolique, etc.).

Pour un processeur, le paramètre important est :

- ✓  $\tau$ , le cycle de base qui inclut le cycle d'accès à sa mémoire locale.
- ✓ Selon ces auteurs le fonctionnement du processeur suit quelques règles (principes) :
- ✓ Le principe de fonctionnement cyclique du processeur cognitif; le système cognitif procède selon un cycle "Reconnaissance Action".
- ✓ Le principe de discrimination : la difficulté de retrouver une information est liée au nombre de candidats répondant aux mêmes indicateurs d'accès.
- ✓ Le principe de rationalité : pour atteindre un objectif, l'individu agit de manière rationnelle. Le comportement rationnel d'un individu résulte de l'union des ensembles : buts à atteindre, structure de la tâche à réaliser, connaissance.
- ✓ La Loi de Fitts : le temps pour placer la main sur une cible dépend uniquement de la précision requise, c'est-à-dire du rapport entre la distance et la dimension de la cible.

### 1.4.3 GOMS

Card, Moran et Newell [1983] présentent aussi le modèle GOMS, qui est un cadre d'analyse de tâche qui permet de décrire ce que sait l'utilisateur sur la tâche et la manière d'effectuer la tâche sans faire de résolution de problème (routine cognitive skills). GOMS permet donc de décrire la connaissance procédurale hiérarchique qu'une personne doit avoir pour exécuter une tâche avec succès. Depuis sa version initiale [Card et al., 1983], différentes versions ont été proposées [John & Kieras, 1996]. GOMS est basé sur l'analyse par application de règle de production. GOMS permet aussi de procéder à des prédictions quantitatives du temps d'exécution d'une tâche particulière.

### 1.4.4 CCT

Kieras et Polson [1985] ont proposé CCT (Cognitive Complexity Theory). C'est une approche formelle pour analyser la complexité des systèmes interactifs du point de vue des utilisateurs. Elle est basée sur des résultats théoriques provenant de la psychologie cognitive. Cette architecture permet de représenter les connaissances des utilisateurs concernant la façon d'utiliser un système pour accomplir des tâches diverses et selon une notation procédurale. Cette notation procédurale permet la quantification de la complexité de la connaissance exigée et la charge de traitement cognitif impliquée en utilisant un système.

Les principales caractéristiques de CCT sont les suivantes :

- La connaissance procédurale nécessaire à l'utilisateur pour exécuter avec succès une tâche est représentée avec des règles de production. Les règles de production adéquates peuvent être générées par une analyse GOMS de la tâche à modéliser.
- Le temps d'exécution de la tâche peut être prédit par un système de production qui interprète les règles accompagnées d'un ensemble de temps d'exécution typiques des opérations.
- Le temps d'apprentissage de la tâche est fonction du nombre de nouvelles tâches que l'utilisateur doit apprendre.
- Quelques prédictions sur l'erreur et la vitesse peuvent être induites des contenus des règles de production,
- Le nombre et le contenu des règles de production reflèteront la complexité de la tâche,

### 1.4.5 Le modèle de Rasmussen

Le modèle de [Card, Moran et Newell,1983] ne prend pas en compte les différentes natures et fonctions des divers processus cognitifs ; tandis que le modèle conceptuel proposé par Rasmussen [1986] prend en considération ces processus. Selon le modèle de Rasmussen, l'entrée perçue de l'extérieur est traitée par divers processus cognitifs travaillant ensemble dans deux modes, un mode conscient et un mode inconscient, pour générer l'action moteur. Plusieurs processus cognitif, tels que le rappel, l'appariement, l'inférence, la prédiction, sont conduits dans la mémoire de travail sous le mécanisme de contrôle de l'attention. C'est comme cela que des relations mutuelles sont construites entre l'information sur le monde extérieur perçu par les organes perceptuels, et les informations, stockées dans la mémoire à long terme, qui sont sélectionnées et activées selon les situations.

Les caractéristiques du modèle de contrôle sont étudiées selon le processus cognitif de l'homme sur des activités de résolution de problème ; c'est-à-dire sur le comportement de l'homme quand il a une tâche à effectuer.

Selon Rasmussen, l'homme conduit la résolution du problème en utilisant la base de connaissance extraite de sa mémoire à long terme à travers un motif assortie avec l'information provenant de l'extérieur. Il existe deux types de règles : les règles symptomatiques (S-Rules) et les règles topographiques (T-Rules) Les S-Rules sont des heuristiques utilisées de façon routinière et qui dépendent de la situation ; elles facilitent les actions des réflexes rapides. Les T-Rules sont des règles générales indépendantes des situations spécifiques ; elles concernent des connaissances abstraites nécessaires au fonctionnement du cerveau. En conclusion, le comportement cognitif de l'homme est classifié en deux modes de contrôle : le mode intentionnel utilisant les T-Rules et le mode schématique utilisant les S-Rules.

L'action de l'opérateur est catégorisée en trois modes :

- mode routinier : à chaque catégorie de problème est associé un type d'action. L'opérateur agit de manière automatique, sans être nécessairement en mesure de justifier techniquement son action.
- mode règle : l'opérateur dispose d'un ensemble de règles du type « si tel diagnostic, engager tel type d'action ».

- mode connaissances sur le fonctionnement du dispositif technique : l'opérateur fait appel à ces connaissances pour trouver comment résoudre le problème, particulièrement dans les situations inhabituelles.

## 1.4.6 Le modèle de Norman

D. Norman dans [Norman, 1986] indique que la réalisation d'une tâche met en jeu au moins sept activités : l'établissement d'un but, la formation d'une intention, la spécification d'une suite d'actions, l'exécution des actions, la perception de l'état du système, l'interprétation de l'état du système, l'évaluation de l'état du système par rapport au but,

Un but est une représentation mentale de l'état à atteindre. L'état d'un système se définit par un ensemble de variables physiques. Un mécanisme de contrôle est un dispositif physique qui permet d'agir sur les variables physiques.

Norman définit la notion de distance qui exprime la dissimilitude entre la représentation de l'image et celle maintenue dans le modèle de l'utilisateur. La distance d'exécution traduit l'effort de mise en correspondance entre la représentation mentale interne de la tâche à effectuer et la représentation physique externe imposée par l'image. La distance d'évaluation traduit l'effort cognitif inverse.

## 1.4.7 Le modèle de Kintsch

Kintsch [1988] postule qu'un modèle cognitivement plausible de la compréhension de textes se doit d'être guidé par les connaissances du lecteur. Les connaissances concernent les mots, la syntaxe, les relations spatiales, le monde, etc. Ces connaissances contraignent et rendent possible la construction des différents niveaux de représentation du discours : structure de surface, micro- et macrostructure sémantiques, modèle de situation.

Le modèle de construction-intégration de Kintsch est un modèle connexionniste hybride. Il associe un système de production symbolique à un processus connexionniste. Le système de production gère une base de connaissance sous forme d'un réseau sémantique. Le modèle comporte deux phases : la phase de construction et la phase d'intégration.

La phase de construction permet de construire une base de texte à partir du texte et de la base de connaissances du sujet. Cette base est formalisée sous forme d'un réseau associatif dont les nœuds représentent des concepts et des propositions. Durant la phase de construction, on suit les étapes suivantes : (a) la formation des concepts et des propositions correspondant directement à l'entrée linguistique ; (b) l'élaboration de chacun de ces éléments par la sélection d'un petit nombre de voisins associés du réseau de connaissances ; (c) l'inférence de certaines propositions supplémentaires ; et (d) l'assignation des forces de connexion à toutes les paires d'éléments qui ont été créées. Le résultat est une base de texte initiale riche, incohérente et contradictoire, qui est alors soumise à un processus d'intégration permettant la formation d'une structure cohérente.

La phase d'intégration permet d'affiner le réseau associatif. Elle permet de renforcer les éléments contextuellement appropriés et inhiber les éléments non reliés ou inappropriés. Elle se déroule de façon itérative jusqu'à la stabilisation du système.

## 1.4.8 Le modèle EXPL

EXPL [Lewis, 1986, 1988] est un modèle mental implémenté sur Prolog et qui est basé sur trois heuristiques. Ces heuristiques sont simples et de bas niveau, mais elles peuvent interpréter la plupart des situations en IHM. Elles indiquent entre autre les relations causales entre les actions utilisateurs, les réponses du système et les buts de l'utilisateur.

Les trois heuristiques sont l'heuristique de l'identité, l'heuristique des extrémités lâches et l'heuristique de l'action précédente. Elles sont successivement décrites ci-après.

### 1.4.8.1 L'heuristique de l'identité

Cette heuristique suggère que si quelque chose apparaît sur l'action de l'utilisateur et sur la réponse du système suivant cette action, alors cette action est probablement une cause de la réponse du système. C'est une spécialisation du principe de similarité de Shultz et Ravinsky [1977] dans lequel les propriétés de cause sont supposées correspondre aux propriétés de leurs effets.

### 1.4.8.2 L'heuristique des extrémités lâches

Selon cette heuristique, si on voit une réponse du système qui contribue au but sur lequel on n'a pas assigné de cause, et qu'on a une action précédente sur laquelle on n'a pas relié un but, alors on peut relier les deux. Cette heuristique combine le principe du déterminisme [Bullock et al., 1982] avec la proposition suivante : toutes les actions utilisateur sont utiles et ont donc des effets qui contribuent au but ; étant donné que le principe du déterminisme stipule que tout événement est supposé avoir une cause. Ainsi, si on perçoit une réponse du système qui contribue au but avec quoi on n'a pas assigné aucune cause (pas français), et qu'on a une précédente action utilisateur qui n'est pas rapportée au but, alors on peut relier les deux ensembles. On peut donc deviner que l'action non expliquée était la cause de la réponse non expliquée.

### 1.4.8.3 L'heuristique de l'action précédente

Si un événement suit une action immédiatement, il est plausible que cette action ait causé l'événement. Cette heuristique combine une forme du principe de coïncidence temporelle [Shultz et Ravinsky, 1977] et du principe de priorité [Bullock et al., 1982]. Selon le principe de coïncidence temporelle, les effets sont supposés être près de leurs causes en temps. Quant au principe de priorité, il stipule que les causes précèdent les effets.

Concevoir des IHM qui respectent ces heuristiques peut conduire à des IHM faciles à apprendre. Ainsi, Polson et al. [1990] se sont inspirés de ce modèle pour construire le modèle CE+.

## 1.4.9 Le modèle CE+

Polson et Lewis [1990] proposent une théorie cognitive de l'apprentissage par exploration d'une IHM, associée à un cadre de conception pour supporter le développement d'un système nécessitant le minimum de connaissance aux utilisateurs. Ce modèle s'appelle apprentissage par exploration CE+.

CE+ comporte un composant de résolution de problème responsable du choix des actions, un composant d'apprentissage qui analyse les effets des choix et mémorise les résultats sous forme de règles CCT, et un composant d'exécution capable d'exécuter les règles résultantes et coordonner l'exécution des règles avec le composant de résolution de problème.

CE+ est issue de CCT [Kieras & Polson, 1985], EXPL [Lewis, 1986] et d'autres théories, d'où son nom CE+ (CCT-EXPL-+). Il incorpore des idées venant de GOMS [Card et al., 1983] et de CCT [Kieras & Polson, 1985] pour la représentation des procédures de connaissances ; il exploite aussi des idées de EXPL [Lewis,1986] et de [Lewis,1988] pour l'apprentissage des procédures à partir des exemples et des recherches sur le processus de résolution de problème.

Le modèle fonctionne comme suit : l'utilisateur choisit une action parmi plusieurs alternatives possibles en se référant à la similarité entre son but et la conséquence attendue de l'action. Après l'exécution de l'action sélectionnée, l'utilisateur évalue la réponse du système en utilisant les heuristiques proposées par Lewis [1986, 1988] ; c'est ainsi qu'il évalue sa progression vers la réalisation de son but. Si le but est réalisé, l'apprentissage se produit en mémorisant l'étape qui vient d'être évaluée sous forme de règle au format CCT. Dans le cas contraire, le composant de résolution de problème est invoqué pour découvrir une action appropriée et ainsi de suite.

### 1.4.10 Le modèle LICAI

Le modèle LICAI est un modèle de l'exploration basée sur le modèle de construction-intégration de Kintsch [Kintsch, 1988]. Il permet de simuler les comportements d'un utilisateur qui doit explorer une interface homme-machine. En effet, selon ce modèle, avant de sélectionner et réaliser une action, l'utilisateur doit comprendre certaines instructions. Ainsi, ce modèle postule les quatre sous-processus suivants :

- Le processus de compréhension des instructions : lorsque les instructions sont données, LICAI crée un réseau de propositions qui codent l'information textuelle donnée. Le réseau comprend un ensemble d'objectifs et de tâches à accomplir ;
- Le processus de sélection du but : LICAI sélectionne un but approprié pour l'action suivante sur la base de la compréhension de l'état courant de la tâche et des informations affichées par le système ;
- Le processus de sélection de l'objet : depuis la mémoire à long terme et sur la base des informations perçues depuis l'interface, les connaissances relatives à la tâche et aux objectifs sont réactivées. Le modèle construit alors un réseau en reliant les connaissances ainsi activées. Les objets de l'interface qui sont les plus saillants (i.e., qui ont le niveau d'activation le plus élevé) sont sélectionnés suite à la diffusion d'activation ;
- Le processus de sélection de l'action : les objets de l'interface sélectionnés sont combinés avec des représentations « génériques » d'actions. Puis, suite à la diffusion d'activation, la représentation générique qui a le plus haut niveau d'activation est sélectionnée.

Le modèle LICAI nécessite donc la prise en compte de la totalité des objets présents sur l'interface lorsqu'une action doit être décidée. Ce qui implique que pour les interfaces qui possèdent plusieurs objets, l'attention des utilisateurs doit au maximum être guidée et orientée vers les parties de l'interface qui sont les plus pertinentes pour les actions à réaliser.

### 1.4.11 Le modèle CoLiDeS

Le modèle LICAI n'est pas approprié pour l'exploration de page web dans la mesure où il y a souvent plusieurs liens et que l'attention des utilisateurs est aussi souvent dispersée en raison des différents éléments de la page. Le modèle CoLiDeS (Comprehension-based Linked model for Deliberate Search) [Blackmon et al., 2000] est une extension du modèle LICAI qui intègre des



processus attentionnels (qui interviennent avant la sélection d'un objet ou d'un lien) et des processus de sélection des actions.

CoLiDeS suppose que la sélection d'une action est un processus à deux phases : la phase d'attention et la phase de sélection. Pendant la première, l'utilisateur segmente la page en un ensemble de zones et génère une brève description de chaque zone à partir de sa connaissance de la convention de structuration de page-écran. L'utilisateur s'occupe de la zone dont la description est perçue comme similaire à son but courant. Pendant la phase de sélection de l'action, l'utilisateur génère les descriptions de tous les outils graphiques dans la zone concernée et agit sur celui dont la description est la plus proche de son but.

### **1.4.12 La conscience de la situation et la représentation occurrente**

Selon Endsley [1995], la conscience de la situation est la perception des éléments d'un environnement dans un volume de temps et d'espace, la compréhension de leur signification et la projection de leur état dans un futur proche. Endsley [1995] définit et propose un modèle de la décision humaine articulé autour de la Conscience de la Situation selon trois niveaux différents : (1) le niveau de la perception des événements, (2) le niveau de l'interprétation de la situation, (3) le niveau de l'anticipation (projection dans le temps concernant l'évolution future de la situation).

Endsley insiste aussi dans son modèle sur le fait que la qualité de la Conscience de la Situation peut-être affectée par la quantité de ressources cognitives disponibles.

Pour Richard [1990], les représentations occurrentes sont des constructions circonstancielles faites dans un contexte particulier et à des fins spécifiques (par opposition aux représentations permanentes, qui correspondent aux connaissances stockées de façon durable en mémoire). A l'image de la conscience de situation, une représentation occurrente constitue un modèle interne de la situation courante.

Hoc et Amalberti [1995] proposent un modèle du diagnostic et de la prise de décision qui s'articule lui aussi autour de la représentation occurrente de la situation. Cette représentation comprend trois facettes : (1) une représentation du processus et des buts à atteindre (qui permet l'anticipation), (2) une représentation des actions possibles (qui sert à orienter le diagnostic vers un niveau de compréhension et une décision adaptée aux moyens disponibles), (2) une représentation des ressources disponibles (qui permet de définir le niveau de compréhension acceptable pour l'action).

Cette représentation occurrente est alimentée en permanence par les informations prélevées sur le processus dynamique (i.e. la situation) et par les connaissances dont dispose l'opérateur en mémoire.

### **1.4.13 Modélisation de l'utilisateur**

On vient de constater qu'il existe plusieurs sources de connaissance concernant les utilisateurs ; Robert [2003] a constaté que ces connaissances viennent de huit sources allant du plus générique au plus spécifique : les études scientifiques, les études statistiques et démographiques, les enquêtes, sondages et études sociologiques, les études de marché, les analyses des utilisateurs, les analyses de tâches, les évaluations ergonomiques et les analyses des problèmes d'utilisation.

La structuration de ces connaissances en vue de pouvoir les exploiter dans un contexte donné s'appelle la modélisation de l'utilisateur. Kobsa et Wahlster [1990] définissent le modèle utilisateur dans le contexte de dialogue en langage naturel avec un système. Selon eux, un modèle utilisateur est une source de connaissance dans un système de dialogue en langage naturel qui contient des suppositions sur tous les aspects de l'utilisateur qui ont un rapport avec le comportement de dialogue avec système.

Razmerita [2003] définit le modèle utilisateur dans le cadre de la personnalisation d'une IHM comme étant une représentation des caractéristiques d'un utilisateur particulier qui peuvent être pertinents pour l'interaction personnalisée. Donc dans le contexte d'évaluation, on peut dire que le modèle utilisateur est l'ensemble des caractéristiques de l'utilisateur qui permettent de prévoir son comportement pendant l'interaction avec le système. En d'autres termes c'est une source de connaissance et des informations sur tous les aspects de l'utilisateur d'un système à évaluer, qui sont pertinents pour l'évaluation du système.

D'après Pohl [1997], le processus de collecte des informations sur les utilisateurs en vue de les mettre à la disposition des systèmes pour adapter leurs comportements ou l'information qu'ils fournissent aux besoins spécifiques des utilisateurs individuels s'appelle modélisation utilisateur. A l'instar de cette définition, nous définissons la modélisation utilisateur comme étant le processus de collecte et d'analyse des informations sur les utilisateurs en vue de les mettre à la disposition des évaluateurs.

Les collectes des informations sur les utilisateurs peuvent être réalisées de plusieurs façons. On peut citer par exemple [Razmerita, 2003] : l'explicitation buts et préférences (par formulaires, verbalisation), la description des informations et caractéristiques personnelles des utilisateurs (par CV), les autoévaluations, les traces des interactions avec le système (log), le eye tracking (oculométrie), les enregistrements vidéo, etc.

Si on possède plusieurs méthodes pour collecter les informations sur les utilisateurs, il faut ensuite identifier les informations pertinentes que doit contenir le modèle de l'utilisateur. Brusilovsky [2007] estime par exemple que les informations suivantes doivent faire partie du modèle :

- La connaissance de l'utilisateur : cette connaissance peut concerner le domaine de l'application, le type d'interface utilisé, l'application ou d'autres applications similaires.
- La motivation, les intérêts et le but de l'utilisateur : ce sont les réponses aux questions : Quoi? et Pourquoi ?
- Les expériences de l'utilisateur : c'est l'ensemble des informations sur l'expérience des utilisateurs sur le domaine de l'application, le type d'interface homme-machine utilisé, l'application ou d'autres applications similaires,
- Les qualités personnelles des utilisateurs : ce sont des caractéristiques personnelles des utilisateurs par exemple s'ils sont plutôt introvertis ou extravertis.
- Le stéréotype des utilisateurs : le stéréotype est un des moyens de modélisation des utilisateurs [Rich, 1979]. Mais Kules [2000] estime que le stéréotype des utilisateurs fait partie des informations que doit contenir le modèle utilisateur au même titre que les connaissances, expériences etc.

## 1.5 Règles ergonomiques

### 1.5.1 Généralités

Les règles ergonomiques désignent des expressions de haut niveau liés à l'ergonomie des logiciels qu'une grande variété d'énoncés de bas niveau qui sont limités à des familles de cas spécifiques [Stephanidis et Akoumianakis, 1999, Mariage et al., 2005]. Les règles ergonomiques peuvent prendre la forme de principes, règles et conventions ou recommandations [Vanderdonck, 1999] :

- Les principes correspondant à des buts et objectifs et objectifs qui guident les décisions conceptuelles durant le cycle de développement. Elles reflètent la large connaissance autour du comportement de l'homme en interaction avec l'ordinateur. Les principes sont généralement exprimés sous forme de directives. Ce sont des objectifs de conception de haut niveau [Marshall et al., 1987].
- Les règles sont basées sur des principes spécifiques pour un champ particulier de conception. Une règle de conception pour le web peut stipuler par exemple d'utiliser une présentation consistante et un langage visuel à travers le site. Les règles sont sujettes à davantage d'interprétation, elles peuvent refléter les besoins d'une organisation particulière ou un cas de conception.
- Les conventions ou recommandations dictent des décisions conceptuelles spécifiques à suivre et devraient refléter les besoins et la terminologie de l'organisation. Ce sont des énoncés non ambigus relatifs à des artefacts physiques.

Beaucoup de règles ergonomiques ont été publiées dans la littérature, on les retrouve dans différentes sources. Mariage [2005] a dressé une catégorisation des règles ergonomiques pour le web. C'est une catégorisation par type (principes, règles, et recommandations) et par source ; on y retrouve les éléments suivants :

- Les règles de conception sont des ensembles de caractéristiques fonctionnelles et/ou opérationnelles qui clarifient la conception d'une interface utilisateur particulière. Leur format permet une exploitation immédiate et ne nécessite aucune interprétation de la part des concepteurs ni des développeurs.
- Les algorithmes ergonomiques traduisent des règles de conception en procédures compréhensibles de telle sorte qu'ils puissent être utilisés plus rapidement. Ils se présentent habituellement sous forme de composant logiciel mettant en application un algorithme.
- Les guides de style comportent un ensemble des directives et/ou de caractéristiques fonctionnelles ou non fonctionnelles visant l'uniformité pour une collection d'interfaces utilisateur distinctes. Cette collection peut être basée sur un système d'exploitation, un éditeur de logiciel, un environnement physique particulier, un domaine d'activité ou une corporation.
- Les normes comportent un ensemble des caractéristiques fonctionnelles et/ou opérationnelles prévues pour normaliser la conception. Les normes sont promulguées par des organismes nationaux ou internationaux de normalisation. Les normes pour l'IHM sont

développés sous l'auspice de l'organisation internationale de la standardisation appelée ISO et de la commission internationale de l'électrotechnique appelée IEC. Elles font l'objet d'une description plus détaillée dans la section suivante.

## 1.5.2 Normes liées à l'IHM

Les normes développées concernant l'IHM peuvent être classées en quatre catégories :

- La qualité d'utilisation de l'artefact : les normes de cette catégorie concernent l'utilisation de l'artefact en termes d'efficacité (*effectiveness*), productivité (*efficiency*) et satisfaction dans un contexte d'utilisation particulier ;
- La qualité de l'artefact : les normes de cette catégorie concernent l'artefact, c'est-à-dire l'interface utilisateur et les interactions qui en découlent ;
- La qualité du processus de développement : les normes de cette catégorie concernent le processus employé pour développer l'artefact et en particulier le processus de développement centré utilisateur ;
- La capacité de l'organisation : les normes de cette catégorie concernent la capacité de l'organisation à appliquer le processus de conception centrée utilisateur,

### 1.5.2.1 Qualité d'utilisation

La norme ISO 9241-11[1998] fournit la définition de l'utilisabilité qui est utilisée dans les autres normes ergonomiques. Selon cette norme, l'utilisabilité est la mesure suivant laquelle un produit peut être employé par des utilisateurs spécifiques réalisant un but spécifique avec efficacité et satisfaction, et de façon productive dans un contexte d'utilisation spécifique.

La norme ISO 9241-11[1998] explique aussi comment identifier l'information dont il faut tenir compte en spécifiant ou en évaluant l'utilisabilité en termes de mesures de performance et de satisfaction des utilisateurs. Des recommandations sont données sur la façon de décrire le contexte de l'utilisation du produit et les mesures de l'utilisabilité d'une manière explicite. Elle comprend aussi une explication de la façon dont l'utilisabilité d'un produit peut être spécifiée et évaluée en tant qu'élément d'un système de qualité. Et enfin, elle explique également comment les mesures de performance et de satisfaction des utilisateurs peuvent être utilisées pour mesurer comment n'importe quel composant d'un système affecte la qualité du système dans sa totalité.

En outre la norme ISO/IEC 9126-1 [2000] décrit les six qualités du logiciel qui sont importantes pendant le développement d'un produit à savoir la fonctionnalité, la fiabilité, l'utilisabilité, l'efficacité, l'entretien et la portabilité. Selon cette norme, l'utilisabilité est la capacité du logiciel à être compris, appris, utilisé par les utilisateurs de façon attractive si ce logiciel est utilisé dans des conditions spécifiées.

Enfin, le rapport technique ISO/IEC DTR 9126-4 [2001] contient des exemples de métriques relatives à l'efficacité, la productivité, la sûreté et la satisfaction.

### 1.5.2.2 Qualité de l'artefact

Cette section concerne les normes qui concernent les artefacts. On décrira les normes qui supportent le développement d'interface utilisateurs. Une partie permet de spécifier des détails sur l'aspect et le comportement de l'interface utilisateur. Une autre partie fournit des recommandations

détaillées sur la conception des interfaces utilisateurs. Et enfin une troisième partie fournit des critères pour l'évaluation des interfaces utilisateur.

### 1.5.2.3 Normes pour la spécification des interfaces homme-machine

Cette section liste les normes concernant la spécification des interfaces homme-machine. Les normes ISO 14915 et IEC 61997 contiennent des recommandations pour les interfaces multimédia. D'autres normes apportent des recommandations spécifiques, à savoir l'ISO/IEC 11581 pour les icônes, l'ISO/IEC 18021 pour les PDA et l'ISO/IEC 10741 pour le contrôle du curseur.

La norme 14915 comprend 4 parties :

**Partie 1 :** Principes de conception et cadre de la norme [2000]. Cette ébauche de norme internationale (Draft International Standard) constitue une introduction globale à la norme.

**Partie 2 :** Contrôle et navigation multimédia [2000]. Cette ébauche (*Committee Draft*) fournit des recommandations pour les structures de navigation et des aides, les contrôles des médias, les contrôles de base, les directives des contrôles de médias pour des médias dynamiques et les contrôles et navigation impliquant de multiples médias.

**Partie 3 :** Combinaison et sélection des médias [2000]. Cette ébauche de norme internationale fournit les orientations pour la combinaison et les sélections des médias, la sélection de médias pour les types d'informations, les combinaisons et l'intégration des médias et l'orientation de l'attention des utilisateurs.

**Partie 4 :** Interface multimédia des domaines spécifiques. Cet article de travail approuvé (*Approved Work Item*) est prévu pour couvrir l'enseignement assisté par ordinateur, le travail coopératif informatisé, les systèmes de kiosque, l'aide en ligne et le test et l'évaluation.

La norme IEC CDV TR 61997 [2000] qui s'intitule « Directives pour les interfaces utilisateurs dans l'équipement multimédia à usage universel » donne des principes généraux et des conseils détaillés de conception pour la sélection de médias, et pour les interfaces utilisateur mécaniques, graphiques et auditifs.

La norme ISO/IEC 11581 qui s'intitule « les symboles et fonctions de l'icône » comporte six parties :

**Partie 1 :** Généralité sur les icônes [2000]. Cette partie contient un cadre pour le développement et la conception des icônes, y compris des conditions générales et des recommandations applicables à toutes les icônes.

**Partie 2 :** Les icônes d'objet [2000]. Cette partie présente les conditions et les recommandations pour les icônes qui représentent des fonctions par association avec un objet, et qui peuvent être déplacées et ouvertes. Elle présente également les spécifications des fonctions et des aspects de 20 icônes.

**Partie 3 :** Les icônes de pointeurs [2000]. Cette partie présente des spécifications et des recommandations pour 8 icônes de pointeurs les plus communément utilisés qui représentent un pointeur lié à un dispositif d'entrée physique. Elle spécifie également comment les icônes de pointeurs changent d'aspect pour fournir un retour d'information aux utilisateurs.

**Partie 4 :** Les icônes de commandes [1999]. Cette partie constitue une ébauche de comité qui contient les spécifications et les recommandations pour 14 icônes de commandes les plus communément utilisées qui permettent à l'utilisateur d'explorer des fenêtres, des listes et d'autres éléments graphiques.

**Partie 5 :** Icônes d'outil [2000]. Cette partie constitue l'ébauche finale de comité (Final Committee Draft) qui contient les spécifications et les recommandations pour 20 icônes d'outil les plus communément utilisés. Elle spécifie aussi les relations entre les outils et les icônes de pointeur.

**Partie 6 :** Les icônes d'action [1999]. Cette partie contient les spécifications et les recommandations pour 23 icônes d'action les plus communément utilisées sur les barres d'outils qui représentent des actions par association avec les objets qui incitent l'utilisateur à rappeler les actions prévues.

La norme ISO/IEC FCD 18021 [2001] qui s'intitule « Technologie de l'information - interface utilisateurs pour les outils mobiles », contient des caractéristiques d'interfaces utilisateur pour PDA avec des possibilités d'échange de données avec les serveurs correspondants.

Enfin la norme ISO/IEC 10741-1 [1995] qui s'intitule « Interaction de dialogue - curseur de commande pour l'édition de textes », spécifie comment le curseur devrait se déplacer sur l'écran en réponse à l'utilisation des touches de commande de curseur.

#### **1.5.2.4 Normes pour la conception des interfaces homme-machine**

Cette section décrit les parties essentielles de la norme ISO 9241 concernant la conception des interfaces homme-machine :

**Partie 10 :** Principes de dialogue [1996]. Cette partie traite des principes ergonomiques généraux qui s'appliquent à la conception des dialogues entre les humains et les systèmes à savoir : (1) l'Adaptation à la tâche, (2) le Caractère auto-descriptif, (3) le Contrôle utilisateur, (4) la Conformité aux attentes utilisateur, (5) la Tolérance à l'erreur, (6) l'Aptitude à l'individualisation, (7) la Facilité d'apprentissage.

**Partie 12 :** Présentation de l'information [1998]. Cette partie contient des recommandations pour présenter et représenter l'information destinées aux visualisations. Elle inclut des conseils sur les manières de représenter l'information complexe utilisant des codes alphanumériques et graphiques/symboliques, la disposition d'écran, ainsi que la conception et l'utilisation des fenêtres.

**Partie 13 :** Soutien aux utilisateurs [1998]. Cette partie fournit des recommandations pour la conception et l'évaluation des attributs des supports aux utilisateurs à savoir les messages d'invite, les retours d'information, les statuts, les aides en ligne et la gestion des erreurs.

**Partie 14 :** Les menus [1997]. Cette partie fournit des recommandations pour la conception des menus utilisés dans des dialogues entre utilisateur et ordinateur. Les recommandations couvrent la structure de menu, la navigation, le choix et l'exécution d'option, ainsi que la présentation de menu par des techniques diverses comprenant le fenêtrage, les panneaux, les boutons, les champs, etc.

**Partie 15 :** Les commandes [1997]. Cette partie fournit des recommandations pour la conception des langages de commande utilisés dans des dialogues entre utilisateurs et ordinateur. Les recommandations couvrent la structure et la syntaxe de langage de commande, les représentations de commande, les considérations des entrées/sorties ainsi que les aides et les retours d'information.

**Partie 16 :** La manipulation directe [1999]. Cette partie fournit des recommandations pour la conception ergonomique des dialogues utilisant la manipulation directe. Elle inclut ainsi la manipulation des objets, et la conception des métaphores, des objets et des attributs. Elle couvre aussi les aspects des interfaces graphiques qui sont manipulés directement, et non couverts par d'autres parties de la série ISO 9241.

**Partie 17 :** Remplissage des formulaires. Cette partie fournit des recommandations pour la conception ergonomique des dialogues se basant sur des remplissages des formulaires. Les recommandations couvrent la structure des formulaires et la considération des entrées/sorties, ainsi que la navigation dans les formulaires.

### **1.5.2.5 Normes relatives aux critères d'évaluation des interfaces homme-machine**

Cette section décrit les parties 2 et 3 de la norme ISO/IEC 9126 :

**Partie 2 :** Les métriques externes [2001]. Ce rapport technique décrit les métriques qui peuvent être utilisées pour spécifier ou évaluer le comportement du logiciel une fois exploité par l'utilisateur. Par exemple : combien de temps nécessite-t-il pour apprendre à employer une fonction, les utilisateurs peuvent-ils annuler des fonctions, les utilisateurs réagissent-ils convenablement aux messages d'erreur ?

**Partie 3 :** Les métriques internes [2001]. Ce rapport technique décrit les métriques qui peuvent être utilisées pour créer les besoins qui décrivent les propriétés statiques de l'interface, c'est-à-dire celles qui peuvent être évaluées par inspection sans exploiter le logiciel. Par exemple : quelle proportion des fonctions est documentée, quelle proportion de fonctions peut être annulée, quels messages d'erreur sont explicites ?

### **1.5.2.6 Normes relatives à la qualité du processus de développement**

Les trois normes suivantes concernent le processus de développement :

- Tout d'abord, la norme ISO 13407 [1999] fournit des conseils sur les activités de la conception centrée utilisateur dans tout le cycle de vie des systèmes informatiques interactifs. C'est un outil pour tous les acteurs concernés par ce processus de conception. Elle fournit des conseils sur les sources d'information et les normes concernant l'approche centrée utilisateur. Elle décrit la conception centrée utilisateur comme une activité multidisciplinaire, qui incorpore des connaissances et des techniques sur les facteurs humains et l'ergonomie, et ceci en vue d'accroître l'efficacité et la productivité, d'améliorer des conditions de travail humaines, et de contrecarrer des effets nuisibles possibles sur la santé humaine lors de l'utilisation, la sûreté et l'exécution. Le processus recommandé est montré dans la figure 1.6.
- Puis l'ISO 16982 décrit les types de méthodes qui peuvent être utilisées. Et enfin, l'ISO/IEC 14598 donne un cadre général pour l'évaluation des logiciels en utilisant le modèle décrit dans ISO/IEC 9126-1.

### **1.5.2.7 Normes relatives à la qualité de l'organisation**

La norme ISO TR 18529 préconise un modèle permettant d'évaluer à quel point une organisation est capable d'effectuer la conception centrée utilisateur. Elle contient un ensemble structuré de processus dérivés de l'ISO 13407.

En outre, la norme ISO/IEC TR 15504 "Technologies de l'information - Évaluation de processus de logiciel", propose un modèle de management de processus du logiciel et fournit une approche structurée pour l'évaluation des processus.

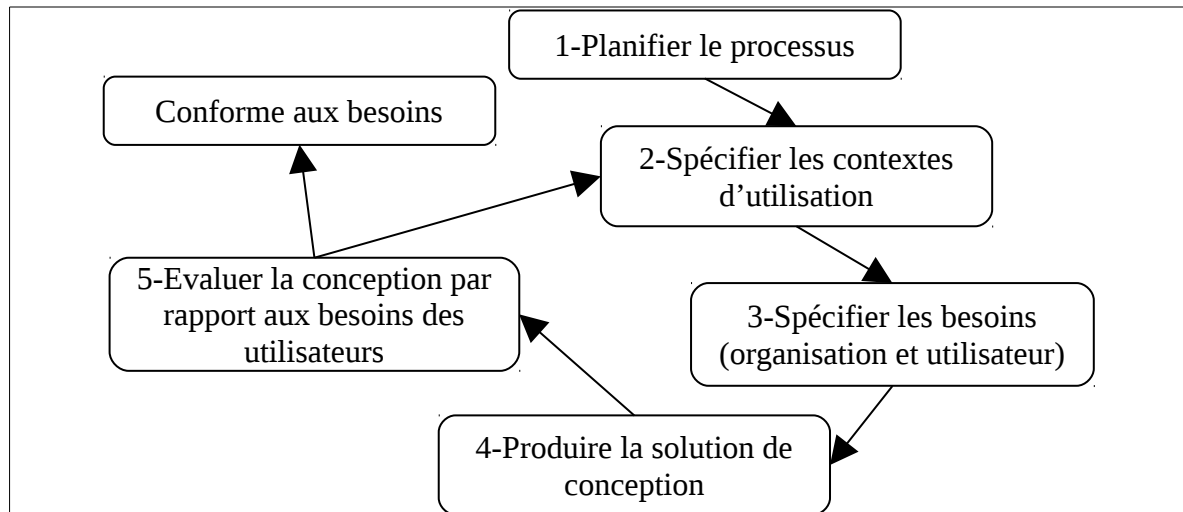


Figure 1.6 : Processus de conception centrée utilisateur

## 1.6 Evaluation des systèmes interactifs : principes et critères

L'évaluation est une étape très importante dans le processus de développement d'un système interactif. Selon Senach [1990], l'évaluation consiste à comparer un modèle de l'objet évalué à un modèle de référence permettant d'établir des conclusions. L'évaluation est réalisée dans le cadre d'un objectif déterminé, et cet objectif détermine les dimensions de l'analyse qui constituent le modèle de l'objet à évaluer. Elle nécessite l'utilisation de méthodes d'évaluation (ou techniques de recueil selon Senach [1990]) Une méthode d'évaluation permet donc de recueillir les données de l'objet évalué et de faire la comparaison avec le modèle de référence. Pour pouvoir faire la comparaison, l'évaluation nécessite aussi des métriques permettant de mesurer les différentes données recueillies.

Les dimensions de l'analyse déterminant les objectifs de l'évaluation peuvent être décomposées. Nielsen [1993] propose la décomposition visible en figure 1.7 pour la dimension acceptabilité du système.

Par ailleurs, Balbo [1994] propose une décomposition en facteurs et critères. Un facteur définit une propriété du logiciel susceptible d'être requise par le client. Il exprime une exigence externe. En revanche le critère est un attribut qui affecte la satisfaction d'un ou plusieurs facteurs. On trouve, par exemple dans la littérature, beaucoup de décomposition du facteur utilisabilité. On peut citer par exemple celle de Abowd, Coutaz et Nigay [1992], dans laquelle l'utilisabilité est un facteur qui peut être mesuré au moyen de trois critères : l'apprentissage, la flexibilité de l'interaction et la robustesse de l'interaction. En revanche, selon Shackel [1991], l'utilisabilité dépend des critères suivants : l'efficacité qui correspond à l'exactitude et la complétude avec lesquelles un utilisateur réalise ses buts, l'efficience qui est l'efficacité du travail accompli en fonction des ressources mobilisées pour obtenir le résultat et enfin la facilité d'apprentissage du système.



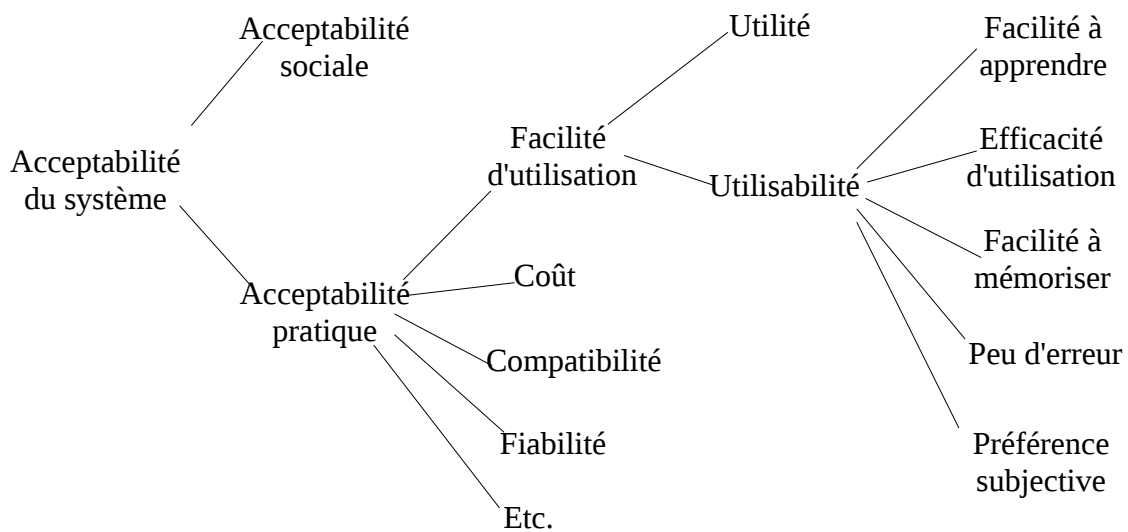


Figure 1.7 : Décomposition de Nielsen [1993]

Dans un premier temps, on va voir de façon plus détaillée les critères de Nielsen [1993] et les critères de Bastien et Scapin [1993] qui sont très reconnus dans le domaine de l'IHM. Ensuite, on discutera des méthodes d'évaluation.

### 1.6.1 Les heuristiques de Nielsen

Nielsen [1993] propose dix critères qui servent de principes généraux pour la conception d'interface utilisateur. Il les appelle « heuristiques » parce qu'ils correspondent plus à des principes de base qu'à des recommandations spécifiques pour l'utilisabilité. Ces heuristiques sont :

1. **Visibilité d'état de fonctionnement du système** : le système devrait toujours tenir les utilisateurs au courant au sujet de ce qui se produit, par l'intermédiaire d'un retour d'information approprié dans un délai raisonnable.
2. **Adéquation du système au monde réel** : le système devrait parler la langue des utilisateurs, avec des mots, des expressions et des concepts familiers à l'utilisateur, plutôt qu'avec des termes orientés système. Il faut respecter les conventions du monde réel, en faisant apparaître les informations dans un ordre naturel et logique.
3. **Contrôle et liberté d'utilisateur** : les utilisateurs choisissent souvent par erreur des fonctions du système et ont besoin d'issues de secours clairement balisées pour quitter la situation non désirée sans devoir parcourir un long dialogue. Il faut leur offrir les fonctions « annuler » et « refaire ».
4. **Cohérence et respect des standards** : les utilisateurs ne devraient pas se demander si différents mots, situations, ou actions signifient la même chose. Il faut se conformer aux conventions de la plate-forme.
5. **Conception minimaliste et esthétique** : les dialogues ne devraient pas contenir des informations qui sont non pertinentes ou rarement utilisées. Toute information superflue dans un dialogue entre en compétition avec les informations pertinentes et diminue la visibilité relative de ces dernières.

6. **Reconnaissance plutôt que rappel** : il faut rendre les objets, les actions et les options visibles. Il ne convient pas que l'utilisateur doive se rappeler des informations données à un endroit du dialogue lorsqu'il est à un autre endroit du dialogue. Les instructions d'utilisation du système devraient être visibles ou faciles à retrouver lorsqu'elles sont nécessaires.
7. **Flexibilité et efficacité d'utilisation** : des raccourcis - non vus par l'utilisateur novice - peuvent souvent accélérer l'interaction pour l'utilisateur expert de telle façon que le système puisse s'adresser à la fois aux novices et aux experts. Il faut permettre aux utilisateurs d'adapter des actions fréquentes.
8. **Aide à la gestion des erreurs**: les messages d'erreur doivent être exprimés dans un langage clair (pas de codes), indiquer précisément le problème et suggérer une solution d'une manière constructive.
9. **Prévention des erreurs** : plutôt qu'un bon message d'erreur, il vaut mieux une conception soignée qui empêche un problème de se produire.
10. **Aide et documentation** : même s'il est préférable qu'un système puisse être utilisé sans documentation, il peut être nécessaire de fournir une aide et une documentation. Toute information de ce type doit être facile à chercher et ciblée sur la tâche de l'utilisateur ; elle doit reprendre la liste des étapes concrètes à suivre et ne doit pas être trop volumineuse.

## 1.6.2 Les critères ergonomiques de Bastien et Scapin

Bastien et Scapin [1993] ont conçu 8 critères ergonomiques pour l'évaluation ergonomique des IHM. Ces critères sont décrits ci-dessous :

Le **guidage** est l'ensemble des moyens mis en œuvre pour prendre en charge l'utilisateur. C'est-à-dire l'ensemble des moyens pour conseiller, orienter, informer, et conduire l'utilisateur lors de ses interactions avec l'ordinateur. Un bon guidage facilite l'apprentissage et l'utilisation du système en permettant à l'utilisateur de comprendre clairement ce qu'il peut faire et comment il peut le faire. Le guidage comprend 4 sous-critères.

Le critère de **incitation** permet de faire connaître à l'utilisateur le contexte dans lequel il se trouve et les actions qu'il peut effectuer. Selon le contexte, on doit faire savoir à l'utilisateur les différentes alternatives et les actions disponibles.

Le critère de **groupement/distinction** permet d'organiser visuellement les différents items les uns par rapport aux autres selon les relations entre les divers items affichés, et leur appartenance ou non à une même classe d'items. Ce critère prend en compte aussi la localisation et le format des items pour exprimer les relations entre eux :

Le critère de **groupement ou distinction par localisation** permet d'éloigner ou rapprocher des items d'un point de vue géographique pour signifier leur différence ou leur ressemblance.

Le critère de **groupement ou distinction par format** permet d'attribuer des formats différents aux items qui sont différents et inversement d'attribuer de format identique aux items qui se ressemblent.

Le critère **retour d'information immédiat** permet d'informer l'utilisateur sur l'action accomplie et sur son résultat. Ce retour d'information doit être fourni dans un délai très court, souvent immédiat.

Le critère de **lisibilité** concerne les caractéristiques matérielles de présentation des informations (typographie, espacement, etc.). Les informations présentées à l'utilisateur doivent être lisibles et compréhensibles.

Le critère **charge de travail** concerne l'ensemble des éléments de l'interface qui ont un rôle dans la réduction de la charge cognitive des utilisateurs et dans l'augmentation de l'efficacité du dialogue. Une charge de travail élevée augmente les risques d'erreurs. Aussi, si l'utilisateur est distrait par des informations non pertinentes, il peut effectuer sa tâche efficacement. Et enfin, les actions requises courtes entraînent des interactions rapides. Plusieurs critères permettent d'alléger la charge de travail de l'utilisateur :

Le critère de **brèveté** permet limiter le travail de l'utilisateur, en perception et mémorisation comme en action. Il permet donc de limiter le travail de lecture et d'entrée d'information. Deux critères participent de la brièveté :

Le critère **concision** concerne les éléments d'entrée/sortie de l'interface qui doivent limiter l'effort à fournir par l'utilisateur. Ce qui veut dire que les informations doivent être présentées de façon claire et adaptée à ce que connaît l'utilisateur dans le monde réel.

Le critère d'**actions minimales** de limiter le nombre d'actions successives nécessaires pour atteindre un but. Pour satisfaire ce critère, on doit éviter de solliciter une action inutile et fournir des raccourcis.

Le critère **densité informationnelle** est un critère qui contribue à limiter la charge de travail de l'utilisateur. Pour satisfaire ce critère, on doit limiter le nombre d'informations présentées en même temps.

Le critère **Contrôle Explicite** concerne à la fois la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions. Quand les utilisateurs définissent et contrôlent eux-mêmes les entrées, les ambiguïtés et les erreurs sont limitées. De plus, le contrôle qu'ont les utilisateurs sur le dialogue est un facteur d'acceptation du système.

Le critère **actions explicites** concerne la relation entre les actions de l'utilisateur et les réponses du système qui doivent être explicites. Lorsqu'un utilisateur est amené à interagir avec un système informatique, il s'attend implicitement à ce que le système lui obéisse. Le système doit exécuter seulement les actions demandées par l'utilisateur au moment où il les demande. On doit donc éviter des actions de la part du système lorsque l'utilisateur n'a rien demandé.

Le critère **contrôle utilisateur** concerne le pouvoir de contrôle de l'utilisateur dans les interactions avec le système. L'utilisateur doit toujours pouvoir contrôler le déroulement des traitements informatiques en cours. En d'autres termes, il doit pouvoir annuler un traitement qu'il aurait demandé ou revenir à une étape antérieure du processus.

Le critère **adaptabilité** concerne la capacité du système à réagir selon le contexte, et selon les besoins et préférences des utilisateurs. Il faut donc fournir à l'utilisateur des procédures, options, et commandes différentes lui permettant d'atteindre un même objectif. Par ailleurs, une interface homme-machine ne peut convenir à la fois à tous ses utilisateurs potentiels. Pour qu'elle n'ait pas d'effets négatifs sur l'utilisateur, cette interface doit, selon les

contextes, s'adapter à l'utilisateur. Le critère d'adaptabilité comprend deux sous-critères : flexibilité et prise en compte de l'expérience de l'utilisateur.

Le critère **flexibilité** réfère aux moyens disponibles à l'utilisateur pour personnaliser son interface de façon à prendre en compte ses stratégies de travail et/ou ses habitudes et les exigences de ses tâches. L'utilisateur doit pouvoir choisir de "personnaliser" l'interface selon sa façon de travailler, les commandes qu'il utilise fréquemment, etc. On doit donc prévoir plusieurs façons d'obtenir le même résultat.

Le critère **prise en compte de l'expérience de l'utilisateur** concerne les différents moyens disponibles pour prendre en compte le niveau d'expérience de l'utilisateur (expérimenté, débutant, occasionnel). Un système qui prend en compte l'expérience de l'utilisateur est susceptible de fournir au novice une aide et à l'expert des raccourcis. Le novice doit être guidé pas à pas dans sa découverte du système alors que l'expert qui est déjà efficace a besoin d'aller plus vite.

Le critère **gestion des erreurs** concerne tous les moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent. La gestion des erreurs est une question fondamentale en informatique. Les interruptions provoquées par les erreurs ont des conséquences négatives sur perturbent l'activité des utilisateurs. De manière générale, elles rallongent les transactions et la planification. Puisqu'il est dans la nature de l'être humain de se tromper, on doit prévoir que l'utilisateur fera des erreurs. On doit surtout concevoir des moyens de pallier ce problème. En d'autres termes, on doit pouvoir : protéger l'utilisateur contre d'éventuelles erreurs, l'avertir lorsqu'il a commis une erreur que l'on peut détecter et enfin, corriger ou l'aider à corriger ses erreurs.

Le critère **protection contre les erreurs** concerne les moyens mis en œuvre pour détecter et prévenir les erreurs d'entrées de données, les erreurs de commandes et les actions à conséquences destructrices. La protection contre les erreurs doit passer par une détection de la part du système. Par exemple, lorsque l'utilisateur est sur le point d'accomplir une action irréversible, le système doit demander une confirmation à ce sujet.

Le critère **qualité des messages d'erreurs** concerne l'expression et la contenu des messages d'erreur à savoir leur pertinence, leur facilité de lecture, leur précision quant à la nature des erreurs, l'indication des actions de correction. Les messages conçus par des techniciens sont souvent exprimés dans un langage technique, qui n'a aucun sens pour un utilisateur grand public. Alors qu'ils devraient être plus contextuels, informer l'utilisateur sur la raison de l'erreur, sur sa nature et sur les façons de la corriger si elles existent.

Le critère **correction des erreurs** concerne les moyens mis en œuvre pour corriger immédiatement les erreurs commises par l'utilisateur. Un système optimisé en termes de correction des erreurs est capable de fournir à l'utilisateur des moyens de faciliter la tâche de correction. Il doit donc guider l'utilisateur concernant les étapes à suivre afin de rectifier l'erreur, tout en fournissant à l'utilisateur les moyens d'identifier l'erreur de façon précise.

Le critère **homogénéité / cohérence** se réfère à la façon avec laquelle les choix de conception de l'interface (codes, dénominations, formats, procédures, etc.) sont conservés pour des contextes identiques, et sont différents pour des contextes différents. C'est un critère qui recouvre la totalité de l'interface et du système sous-jacent. Les procédures,

labels, commandes, etc., sont d'autant mieux reconnus, localisés et utilisés, que leur format, localisation, ou syntaxe sont stables d'un écran à l'autre, d'une session à l'autre. Dans ces conditions le système est davantage prévisible et les apprentissages plus généralisables ; les erreurs sont réduites. Le manque d'homogénéité peut augmenter considérablement le temps de recherche. Le manque d'homogénéité est aussi une raison importante de refus d'utilisation.

Le critère **signifiante des codes et dénominations** concerne l'adéquation entre l'objet ou l'information affichée ou entrée, et son référent. Des codes et dénominations signifiants disposent d'une relation sémantique forte avec leur référent. La réflexion sur les codes et dénominations devra prendre en compte les standards existants dans des interfaces comparables, afin de capitaliser sur les connaissances préalables des utilisateurs. Pour les applications professionnelles ou spécialisées, il sera essentiel de s'imprégner du vocabulaire des personnes afin de concevoir une interface signifiante pour eux.

Le critère de **compatibilité** est très large. Il se réfère à l'accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, habitudes, compétences, âge, attentes, etc.) et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part. De plus, la compatibilité concerne également le degré de similitude entre divers environnements ou applications.

### 1.6.3 Méthodes d'évaluation des systèmes interactifs

Selon le méta-modèle de Senach [1990], l'évaluation nécessite des méthodes adaptées (techniques de recueil de données). Il existe dans la littérature de nombreuses méthodes d'évaluation. Les praticiens ont donc l'embarras de choix ; ce qui amène certains chercheurs à faire des classifications pour aider les praticiens à choisir. A titre d'exemple, Calvary et Coutaz [1999] proposent une classification selon le caractère sommatif ou formatif de la méthode. Balbo [1994], quant à elle, propose une classification selon cinq dimensions à savoir, les connaissances, les ressources matérielles, les facteurs situationnels, les moyens humains et les résultats. Grislin et Kolski [1996] proposent une classification selon sept critères : les variables cibles, les intervenants de l'évaluation, les types de connaissances, les types de méthodes, l'existence ou non de l'interface, les contraintes d'application de la méthode et le positionnement dans le cycle de développement du système<sup>3</sup>. Dans le cadre de ce mémoire, on se focalise essentiellement sur les évaluations de type test utilisateur ainsi que sur les méthodes d'inspection. On va voir dans ce chapitre la méthode « test d'utilisabilité » qui est une méthode du type test utilisateur puis les méthodes heuristiques.

#### 1.6.3.1 Les tests d'utilisabilité

Les tests d'utilisabilité font parties des méthodes d'évaluation les plus populaires. Selon Dumas et Fox [2008], cette méthode présente six caractéristiques :

- L'évaluation se focalise sur l'utilisabilité : ce qui signifie que les objectifs de l'évaluation doivent concerner l'utilisabilité. Ce qui n'est pas toujours le cas selon Dumas et Fox (2008) : on a souvent tendance à ajouter d'autres questions telles que des questions concernant le marketing.

---

<sup>3</sup> Le lecteur intéressé par des ouvrages généraux, des chapitres ou des articles de synthèse concernant l'évaluation des systèmes interactifs consultera par exemple [Helander et al., 1997] et [Jacko et Sears, 2002].

- Les participants sont des utilisateurs finaux ou des utilisateurs potentiels : tester une IHM avec des autres participants que les utilisateurs finaux permet de détecter des problèmes. Par contre, les résultats ne peuvent pas être généralisés sur les vrais utilisateurs finaux ou les vrais utilisateurs potentiels.
- Le produit ou l'IHM à évaluer existe : il peut correspondre à des schémas de conception d'interface utilisateur, des premiers prototypes de bas niveau, des prototypes fonctionnels de haute fidélité, des produits en bêta version ou des produits déjà réalisés
- Les participants exécutent des tâches et souvent en pensant à haute voix : durant la phase d'exécution de l'évaluation, les évaluateurs et l'administrateur interagissent. Au début de la session, l'administrateur instruit les participants sur le processus d'évaluation tout en soulignant que l'évaluation concerne l'utilisabilité.
- Les données sont enregistrées et analysés : l'enregistrement utilise parfois un enregistreur vidéo ; on peut aussi se baser sur un rapport d'exploitation du système ou sur des notes utilisateur crayon/papier ou informatisées.
- Les résultats de l'évaluation sont communiqués aux audiences appropriés : La communication de ces résultats peut être informelle. Le rapport formel sert à documenter les processus plutôt qu'à diffuser les résultats.

Cette méthode a l'avantage d'être simple. Elle nécessite peu d'expertise, et peut fournir un aperçu utile sur les problèmes liés au système interactif. Toutefois, il existe un certain nombre de problèmes. L'information est nécessairement subjective, et peut être sélective en fonction des tâches choisies. Mesurer les degrés de sévérité des problèmes est souvent difficile. Parfois on utilise l'oculométrie, technique permettant d'enregistrer les mouvements oculaires [Duckovski, 2003]. L'analyse des mouvements oculaire peut aider à identifier les sources exactes de confusions ou erreurs.

### **1.6.3.2 Les méthodes d'inspection de l'utilisabilité**

Les méthodes d'inspection de l'utilisabilité [Nielsen & Mack, 1994] demeurent les plus importantes méthodes à bas coût pour l'évaluation de l'utilisabilité. Elles peuvent être appliquées à toute IHM tout au long de son développement : un prototype papier, un prototype fonctionnel, un logiciel en test de production, ou une version diffusée et installée. Ce sont des méthodes d'évaluation analytique, qui ne nécessitent pas la présence des utilisateurs finaux ou des utilisateurs potentiels, contrairement aux méthodes empiriques telles que des tests d'utilisabilité. Parmi les méthodes d'inspection les plus connues, on retrouve l'évaluation heuristique et la méthode cognitive walkthrough.

L'évaluation heuristique utilise une liste d'heuristiques pour identifier et confirmer des problèmes d'utilisabilité. Il existe plusieurs listes d'heuristiques exploitables. Basée sur les heuristiques de Nielsen (cf. 1.6.1), on trouve la méthode d'évaluation heuristique de Nielsen [Nielsen, 1994]. Basée sur les critères de la norme ISO 9241, la méthode ISO 9241 Evaluator a été proposée par [Reitterer et Oppermann, 1995 ; Oppermann et Reitterer, 1997] ; de même IsoMetrics a été proposée par [Gediga, Hamborg, et Düntsch, 1999]. On peut aussi utiliser les critères

ergonomiques de Bastien et Scapin [1993] en support à une méthode d'évaluation par des critères ergonomiques [Bastien et Scapin, 1995].

L'évaluation est en général composée de deux phases. Tout d'abord, plusieurs évaluateurs analysent indépendamment l'IHM en utilisant les heuristiques ou critères. Puis les problèmes identifiés sont rassemblés dans une liste unique puis discutés en groupe. Une session d'évaluation dure en générale une à deux heures. On la fait souvent en deux sessions : la première pour avoir une impression d'ensemble de l'interface et la deuxième pour se concentrer sur des éléments spécifiques de celle-ci. Le rapport d'évaluation se présente comme une liste d'éléments défectueux de l'interface, avec pour chaque élément la description des problèmes, avec le plus souvent des références précises aux heuristiques.

Cognitive Walkthrough vise à mettre en évidence les problèmes d'utilisabilité rencontrés par un utilisateur pendant l'accomplissement d'une tâche donnée. Elle n'implique pas les utilisateurs de la tâche à évaluer. C'est l'évaluateur dit expert qui explore l'interface en essayant d'agir comme les utilisateurs réels de la tâche. La méthode dispose d'une série de questions auxquelles l'évaluateur doit répondre pour l'aider à détecter les éventuels problèmes d'utilisabilité. La méthode cognitive walkthrough a beaucoup évolué depuis sa proposition en 1990 par Lewis et al. [1990] et dispose de plusieurs versions et variantes actuellement [Mahatody et al., 2007a, 2007b]. Un chapitre entier lui est consacré (le chapitre 2).

## **1.7 Conclusion**

On a vu que pour maîtriser la conception et l'évaluation d'IHM, qui est une science pluridisciplinaire, de nombreux chercheurs et praticiens tentent de produire et valider des principes de base, des modèles, méthodes, guides et recommandations divers à ce sujet. On a vu les principes de la conception centrée utilisateur. De ces principes découlent plusieurs normes liées à l'IHM. Ils sont répartis en quatre catégories selon qu'ils concernent la qualité d'utilisation du système interactif, la qualité du système interactif lui-même, la qualité du processus de développement du système ou la qualité de l'organisation qui conçoit le système.

On a vu aussi dans ce chapitre quelques heuristiques et normes parmi les plus reconnues concernant les IHM : il s'agit des heuristiques de Nielsen et des critères ergonomiques de Bastien et Scapin. De ces heuristiques et normes découlent des méthodes d'évaluation.

D'autres chercheurs tentent de développer et fournir des théories visant à faciliter la modélisation des systèmes interactifs ainsi que leurs différents composants. On a vu ainsi quelques théories et modèles sur l'artefact, les tâches et les utilisateurs. Sur les tâches par exemple, cinq modèles de tâches ont été parcourus. Sur les utilisateurs, on a vu plusieurs théories et modèles les concernant dont le modèle CE+ d'apprentissage par exploration et le modèle de Norman, ces deux modèles étant à la base de la méthode d'évaluation Cognitive Walkthrough.

On a souligné l'importance de l'évaluation dans le processus de développement d'un système interactif et présenté deux types de méthodes d'évaluation : les tests d'utilisabilité et l'inspection de l'utilisabilité. La méthode Cognitive Walkthrough fait partie de la deuxième catégorie de méthode d'évaluation, et elle fait l'objet d'une étude détaillée dans le chapitre suivant.

# Chapitre 2 - La méthode Cognitive Walkthrough et ses différentes versions et variantes

---

## Plan du chapitre

2.1- Introduction.....	2
2.2- Evolution des versions originales.....	2
2.2.1 - Première version.....	3
2.2.2 - Deuxième version.....	3
2.2.3 - Troisième version.....	4
2.3- Les autres variantes.....	5
2.3.1 - Heuristic Walkthrough (HW).....	5
2.3.2 - Norman's Cognitive Walkthrough.....	5
2.3.3 - Streamlined Cognitive Walkthrough.....	7
2.3.4 - Cognitive Walkthrough for the Web.....	7
2.3.5 - Groupware Walkthrough.....	9
2.3.6 - Activity Walkthrough.....	10
2.3.7 - Interaction Walkthrough.....	10
2.3.8 - Cognitive Walkthrough with users.....	12
2.3.9 - Extended Cognitive Walkthrough.....	12
Chapitre 3- Distributed Cognitive Walkthrough.....	13
3.1.1 - Enhanced Cognitive Walkthrough.....	14
3.2- Synthèses.....	15
3.2.1 - Synthèses selon les aspects conceptuels, méthodologiques et technologiques.....	15

---



## 2.1 Introduction

On a mentionné dans le chapitre précédent que le développement d'un système interactif est un processus le plus souvent itératif ; l'évaluation est un composant important de ce processus et fait l'objet de nombreuses études et recherches depuis de nombreuses années par la communauté scientifique en Interaction Homme Machine (IHM). Il existe aujourd'hui de nombreuses méthodes et techniques d'évaluation ; plusieurs études ont été faites proposant des revues et synthèses de celles-ci. Citons par exemple : [Dix et al; 1993], [Sweeney et al., 1993], [Grislin et Kolski, 1996] et [Sears, 2003]. L'inspection de l'utilisabilité est un nom générique d'un ensemble de méthodes basées sur le fait que un ou plusieurs évaluateurs inspectent ou examinent les aspects de l'utilisabilité de l'interface utilisateur [Mack et Nielsen, 1994] [Virzi, 1997] [Cockton et al., 2003] [Bastien, 2004]. Selon Nielsen et Mack [1994] (page 3), l'utilisabilité est un concept assez large qui se rapporte à l'apprentissage et à l'utilisation du système par l'utilisateur, de même qu'à la satisfaction de l'utilisateur utilisant le système.

Ce chapitre est consacré à la méthode Cognitive walkthrough et à un ensemble de ses versions et variantes. Il a été traduit en anglais et a fait l'objet d'une publication dans une revue [Mahatody et al., 2010]. C'est un document de synthèse qui concerne l'évaluation des systèmes interactifs sous l'angle des méthodes d'inspection et plus particulièrement de la méthode Cognitive Walkthrough (CW). Les principes de base de CW sont abordés en partant de ses trois principales évolutions. Puis onze variantes significatives de CW sont passées en revue. Plusieurs synthèses sont proposées : la première selon les points de vue conceptuel, méthodologique et technologique, deux autres synthèses selon les études comparatives et non comparatives existantes, la dernière en vue d'une aide au choix d'une version ou variante.

Cognitive Walkthrough (CW) fait partie des méthodes d'inspection de l'utilisabilité. C'est une méthode qui associe le parcours (walkthrough) de l'interface avec un modèle cognitif. L'évaluateur parcourt l'interface dans le contexte des tâches qu'un utilisateur typique de l'interface aura besoin d'accomplir. Les actions et les réponses de l'interface sont évaluées par rapport aux buts et connaissances de l'utilisateur, en répondant à des questions se rapportant au modèle cognitif de la méthode ; les écarts entre les attentes de l'utilisateur et les étapes requises par l'interface sont notés. Comme les autres méthodes d'évaluations les plus courantes en IHM, CW met l'accent sur les principes de base de l'utilisabilité ; mais à la différence des autres méthodes d'évaluation, CW se concentre aussi sur les activités cognitives de l'utilisateur, spécialement sur ses buts et ses connaissances en exécutant une tâche spécifique.

CW a été proposée par Lewis et al. [1990]. Mais depuis, elle a beaucoup évolué. En effet, des modifications ont été faites ensuite et des variantes ont été aussi proposées par d'autres auteurs. Le principe reste le même, la méthode exploite un modèle cognitif qui va être simulé en répondant à des questionnaires se rapportant à ce modèle. En nous inspirant de Mahatody et al. [2007a et 2007b], nous allons décrire dans un premier temps l'évolution de la méthode initiale qui se résume à trois versions : celle de Lewis et al. [1990], la seconde de Polson et al. [1992], et enfin la troisième de Wharton et al. [1994]. Ensuite nous passerons en revue différentes variantes significatives de CW : Heuristic Walkthrough, Norman's CW, Streamlined Walkthrough, CW for the Web, Groupware Walkthrough, Activity Walkthrough, Interaction Walkthrough, CW with users, Extended CW, Distributed Cognitive Walkthrough et Enhanced Cognitive Walkthrough. Enfin, plusieurs synthèses seront proposées en fin de chapitre.

## 2.2 Evolution des versions originales

### 2.2.1 Première version

La première version de Lewis et al [1990] a été destinée à l'évaluation des interfaces de type « walk-up-and-use », c'est-à-dire qu'on utilise sans ou avec peu de formation initiale. Elle est basée sur la théorie de l'apprentissage par exploration CE+ [Polson et Lewis, 1990]. Le modèle CE+ est une théorie cognitive d'apprentissage de l'utilisation pour la première fois d'une IHM. Des guides de conception appelés « Design Principles for Successful Guessing » sont dérivés de cette théorie pour supporter la conception de système interactif nécessitant peu ou éventuellement pas de formation utilisateur. Le modèle CE+ est composé de trois composants principaux : un composant de résolution de problème, un d'apprentissage et un d'exécution. Le modèle fonctionne comme suit : l'utilisateur du système choisit une action parmi plusieurs alternatives possibles en se référant à la similarité entre son but et la conséquence attendue de l'action ; après l'exécution de l'action sélectionnée, l'utilisateur évalue la réponse du système en utilisant les heuristiques proposées par Lewis [Lewis, 1986, 1988] ; c'est ainsi qu'il évalue sa progression vers la réalisation de son but. Si le but est réalisé, l'apprentissage se produit en mémorisant l'étape qui vient d'être évaluée sous forme de règle selon le format proposé par Kieras [1985]. Dans le cas contraire, le composant de résolution de problème est invoqué pour découvrir une action appropriée et ainsi de suite.

Pour cette première version de CW, l'évaluation se fait en deux phases : d'abord, l'évaluateur ou groupe d'évaluateur spécifie une série de tâches à évaluer. Il est à noter que l'évaluateur peut être le concepteur, l'utilisateur ou l'expert en inspection d'utilisabilité. Ensuite il faut décomposer chaque tâche en séquence d'actions (phase de préparation). Finalement l'inspection de chaque action est abordée en répondant aux questions relatives au processus de simulation du modèle CE+ décrits plus haut : choix et exécution de l'action puis évaluation de la réponse du système (phase d'évaluation). Lewis et al [1990] constatent que la méthode est prometteuse mais n'est pas encore satisfaisante. En effet selon ces auteurs il n'y a que 50% d'erreurs observées qui sont identifiées par cette méthode.

### 2.2.2 Deuxième version

Pour améliorer la première version, une deuxième version est proposée par Polson et al [1992]. Elle est basée sur une extension du modèle CE+. Le nouveau modèle de cette version est lié à la théorie de l'action de Norman [Norman, 1986] et au modèle de construction-intégration de Kintsch [Kintsch, 1988] :

1. La théorie de Norman dit que la réalisation d'une tâche met en jeu sept étapes : l'établissement d'un but, la formulation d'une intention, la spécification d'une suite d'actions, l'exécution des actions, la perception et l'interprétation de l'état du système et enfin l'évaluation de l'état du système par rapport au but considéré.
2. Le modèle de Kintsch est un cadre conceptuel permettant de construire une structure connexionniste en deux phases. Dans la première phase de construction toutes les connexions ou relations possibles sont construites. Pendant la deuxième phase dite d'intégration, les connexions ou relations inadéquates sont éliminées en utilisant les contextes du domaine étudié. Ici le modèle de Kintsch est utilisé pour intégrer la représentation des textes ou des objets perceptuels de l'interface en tenant compte de l'expérience et des connaissances de l'utilisateur.

Les caractéristiques de l'utilisateur, le contexte et l'environnement d'utilisation du système sont pris en compte. Selon le nouveau modèle intégré à cette seconde version, pour exécuter une

tâche, l'utilisateur doit construire une structure de buts. Une structure de but est similaire à la structure hiérarchique de GOMS [Card et al., 1983] : le but principal est décomposé en buts intermédiaires correspondant à la décomposition de la tâche, et les buts finaux correspondent aux actions. Cette structure de but est construite à partir du but initial en y intégrant c'est-à-dire y liant les propositions représentant ses connaissances de base, les propositions représentant les objets qu'il aperçoit dans les environnements et les actions. Une action est activée (exécutable) si des liaisons existent entre le but initial et cette action. Après l'exécution de l'action, chaque réponse du système est interprétée entraînant la désactivation du but atteint et la construction de nouvelles propositions. Ces nouvelles propositions sont liées à la structure actuelle entraînant de nouvelles activations d'actions, et ainsi de suite.

La méthode se déroule en deux phases : la préparation et l'évaluation. La phase de préparation consiste à sélectionner les tâches à évaluer, les décomposer en séquences d'actions et déterminer les buts de l'utilisateur. La phase d'évaluation est guidée par trois formulaires. Les éléments des formulaires traitent les problèmes et les échecs qui peuvent se produire lors du processus d'exploration du modèle. Le premier formulaire traite de la relation entre les buts nécessaires pour manipuler l'interface et les buts courants de l'utilisateur. Le deuxième s'occupe des problèmes liés à la sélection de l'action en supposant que les buts sont appropriés. Et enfin le troisième formulaire concerne l'évolution des buts de l'utilisateur en prenant en compte la réponse du système après l'exécution de l'action.

Cette version devient plus complexe et lourde à utiliser (Polson et al., 1992). En effet, l'évaluateur doit faire des analyses extrêmement détaillées. Il doit décrire explicitement les buts et actions des utilisateurs, donner une analyse détaillée sur la façon dont l'utilisateur choisira une action et enfin décrire comment le retour d'information et son interprétation par l'utilisateur changeront les buts.

Pour pallier la complexité et à la lourdeur de cette version de CW, Rieman et al. [1991] proposent une version automatisée implémentée sous Hypercard d'Apple. Cette version permet à l'évaluateur de remplir les formulaires sans utiliser de fiches papier. Elle fournit aussi un éditeur simple pour gérer les structures des buts, de même que des aides pour chaque question.

### **2.2.3 Troisième version**

La version précédente a fait l'objet des différentes études et critiques [Rowley et Rhoades, 1992 ; Wharton et al., 1992 ; Desurvire et al., 1992 ; Cuomo et Bowen, 1992]. Ces études confirment la lourdeur de la méthode. Et même avec l'aide de l'automatisation, elle reste excessivement coûteuse en temps. Une nouvelle version est développée par Wharton et ses collègues pour faire face à ces critiques.

Pour cette version, l'évaluateur est invité à imaginer un scénario spécifique et crédible pour chaque action que l'utilisateur doit exécuter pour accomplir sa tâche. Pour rendre le scénario crédible, l'évaluateur doit justifier ou critiquer chaque action en tenant compte des connaissances de base de l'utilisateur, et des réactions de l'interface. Pour l'aider à évaluer chaque action de la tâche, l'évaluateur doit répondre à quatre questions relatives respectivement à ce que pense l'utilisateur au début de l'action (Q1 : Will the user try to achieve right effect ?), à sa capacité à localiser la commande (Q2 : Will the user notice that the correct action is available ?), à sa capacité à identifier la commande (Q3 : Will the user associate the correct action with the effect that user is trying to achieve ?) et enfin à sa capacité à interpréter le retour d'information (Q4 : If the correct action is performed, will the user see that progress is being made toward solution of the task ?).

Des études critiques ont été aussi faites au sujet de cette troisième version [John et Packer, 1995 ; Sears et Hess, 1999 ; Jacobsen et John, 2000 ; Riihihoo, 2000 ; Huart et al., 2004]. La plupart des auteurs s'accordent sur le fait que CW est fastidieuse même si elle est facile à apprendre et à utiliser.

## 2.3 Les autres variantes

En raison de son intérêt pratique (tout en s'appuyant sur une base théorique), la méthode CW a évolué. Cette évolution a visé à compenser certaines limites, prendre en compte certaines avancées dans le domaine de l'interaction homme-machine, ou types d'IHM particuliers, tels que le web, le multimédia, etc. [Jeffrey et al., 1991 ; John et Packer, 1995 ; Jacobsen et John, 2000 ; Huart et al., 2004], Dans cette partie, nous passons en revue chronologiquement onze évolutions de CW.

### 2.3.1 Heuristic Walkthrough (HW)

Heuristic Walkthrough [Sears, 1997] est une méthode d'évaluation composée de deux processus d'inspection comme dans la méthode Usability Walkthrough [Karat et al., 1992] : un processus d'inspection à base de scénario, et un processus d'inspection à base d'heuristiques.

Heuristic Walkthrough est donc une combinaison de la méthode Heuristic Evaluation [Nielsen, 1992 ; Nielsen et Molich, 1990] et de CW [Wharton et al., 1994]. Pour Heuristic Walkthrough, l'évaluation se fait en deux phases. La phase 1 est une évaluation guidée par une liste priorisée de tâches, et une liste de questions provenant de CW. Pendant la phase 2, l'évaluateur est libre d'explorer n'importe quel aspect du système en l'utilisant. Néanmoins, il est guidé par la connaissance obtenue dans la phase 1, la liste des tâches, une liste d'heuristiques (par exemple celles fournies dans [Nielsen, 1992 ; Nielsen et Molich, 1990]) et la liste de questions utilisées dans la phase 1.

Sears a utilisé les critères proposés par [Bastien et Scapin, 1995] pour comparer les trois méthodes : Heuristic Evaluation, Cognitive Walkthrough et Heuristic Walkthrough. Ces critères sont : validity (capacité de la méthode à focaliser sur les aspects spécifiques et pertinents de l'interface), thoroughness (capacité de la méthode à évaluer tous les aspects de l'interface), reliability (capacité de la méthode à produire le même résultat sous une même condition). Sears a démontré que HW permet de trouver plus de problèmes d'utilisabilité que Cognitive Walkthrough. En plus elle produit moins de faux problème d'utilisabilité que Heuristic Evaluation.

### 2.3.2 Norman's Cognitive Walkthrough

Norman Cognitive Walkthrough est issue du projet AVANTI (AdaptiVe and Adaptable iNteractions for multimedia Telecommunication applIcation) pour faire face aux spécificités du projet (éloignement des équipes de concepteurs dans différentes villes, voire différents pays) ainsi que pour positionner des problèmes d'interaction à un haut niveau d'abstraction [Rizzo et al., 1997]. En effet, dans ses versions initiales, CW n'est pas particulièrement adéquate pour des équipes de concepteurs composées de personnes de différentes cultures. En plus, CW se concentre sur les interactions de bas niveau (frappe au clavier, clic de souris...).

Norman's CW est certes basé sur le modèle de Norman [1986], mais celui-ci est modifié par [Rizzo et al., 1997] selon la figure 2.1. Selon Hutchins et al. [1985], une distance cognitive indique la quantité et la qualité de traitement d'information nécessaire pour remplir l'espace entre les états (intention, évaluation, etc.). La notion de distance cognitive peut être à la fois liée à l'exécution de l'action et aussi à l'évaluation du résultat. Dans le premier cas elle se rapporte à la quantité de traitement de l'information nécessaire pour réduire le fossé entre l'intention et l'action

physique sur laquelle l'intention est communiquée au système. En d'autres termes, elle se rapporte à l'acte de traduction des pensées et des buts de l'utilisateur dans le langage du système. Dans le dernier cas, la distance cognitive se rapporte à l'effort mental nécessaire à la traduction de l'information affichée par le système dans les termes du modèle conceptuel adopté par l'utilisateur. L'évaluation consiste à mettre à l'épreuve les activités de l'utilisateur exécutant une ou plusieurs tâches à l'intérieur d'un scénario donné. L'évaluateur explore le système en cherchant les actions qui doivent contribuer à l'exécution de la tâche. Il choisit des actions dont la description correspond avec ce qu'il essaie de faire. Ensuite il interprète la réponse du système en vue d'évaluer si on progresse vers l'achèvement de la tâche ou s'il faut reconsidérer le but. Ce qui permet donc d'identifier d'une part si la signification et la forme de l'interface sont bien interprétées par l'utilisateur et d'autre part s'il est capable de fixer un but faisable et d'exécuter l'action correcte sur l'objet adéquat.

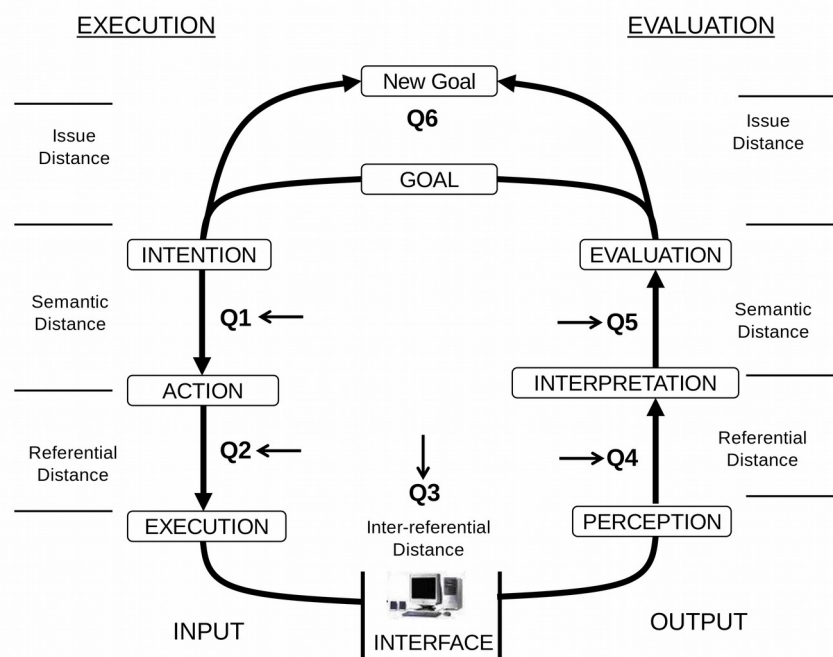


Figure 2.1 : Le modèle de modifié de Norman adapté de [Rizzo et al., 1997]

A chaque étape de l'interaction, des questions doivent être renseignées :

- Q1 : Est-ce que l'action correcte et faisable est suffisamment mise en évidence pour l'utilisateur et est-ce que les actions correspondent avec l'intention exprimée par l'utilisateur ?
- Q2 : Est-ce que l'utilisateur relie la description de l'action correcte avec ce qu'il essaie de faire ?
- Q3 : Est-ce que l'utilisateur reçoit le retour d'information au même endroit et avec la même modalité suite à l'exécution de l'action ?
- Q4 : Est-ce que l'utilisateur interprète correctement la réponse du système sur l'action choisie c'est-à-dire est-ce qu'il réalise qu'il a fait le bon ou mauvais choix ?
- Q5 : Est-ce que l'utilisateur évalue correctement les résultats c'est-à-dire qu'il est capable d'évaluer s'il s'approche du but ?

11. Q6 : Si le but n'est pas bon (ou peut être amélioré), est-ce qu'il comprend que l'intention qu'il essaie d'exécuter ne peut pas être accomplie dans l'état actuel (ou est-ce qu'il trouve de buts alternatifs pour l'améliorer) ?

Si la réponse n'est pas complètement affirmative, l'évaluateur communique la question aux membres de l'équipe, avec la spécification des solutions alternatives. Ce qui permet aux membres de l'équipe répartis dans différentes localités de communiquer sans ambiguïté.

### 2.3.3 Streamlined Cognitive Walkthrough

Spencer [2000] constate que la troisième version de CW est difficile à appliquer dans de grandes compagnies de développement de logiciel, qu'elle ne produit pas systématiquement de bons résultats et que l'équipe de développement peut l'estimer inutile. Selon lui, la difficulté viendrait de plusieurs contraintes :

- La pression temporelle est une contrainte majeure pour les concepteurs. Ils sont souvent sous pression pour terminer à temps leurs missions, et toute activité non immédiatement utile et nécessitant beaucoup de temps est abandonnée. C'est le cas de la troisième version de CW qui nécessite beaucoup de temps pour répondre aux quatre questions (cf. §2.3). En plus elle produit des résultats volumineux et demande du temps supplémentaire pour les traiter.
- La longue discussion de conception qui apparaît lors de l'identification de problèmes pendant la session d'évaluation. En effet, l'équipe d'évaluation tente de résoudre les problèmes. Ce qui fait que le temps d'évaluation augmente ou est dépensé pour la conception.
- La défense des solutions prises pour la conception est une attitude que certains concepteurs ont pendant la session d'évaluation : ils pensent qu'ils ont déjà consacré beaucoup de temps à concevoir le système et tentent donc de défendre la conception faite afin d'éviter de consacrer du temps de travail supplémentaire au projet si les critiques évoquées sont prises en compte.

Pour faire face à ces contraintes, Spencer propose Streamlined CW. C'est une variante de la 3ème version de CW ; elle comporte 5 phases. La première est la phase de définition de l'entrée du parcours d'inspection (walkthrough). La deuxième consiste en une réunion des évaluateurs, pour définir ce qu'il faudra faire et ce qu'il ne faudra pas faire pendant l'évaluation, ce qu'il faudra éviter pour que les concepteurs ne soient pas sur la défensive vis-à-vis de leur conception, pour définir le rôle de chaque membre de l'équipe. Pendant la phase d'inspection (troisième phase), l'évaluateur procède de la même manière que dans CW en répondant aux deux questions suivantes :

- ◆ Est-ce que l'utilisateur sait ce qu'il doit faire à cette étape ?
- ◆ Si l'utilisateur réalise correctement l'action, est ce qu'il saura qu'il a réalisé la bonne action et qu'il progresse vers le but en question ?

La quatrième phase consiste à enregistrer les problèmes détectés. Et enfin la dernière consiste à les corriger. Cette méthode insiste beaucoup sur la rigueur au niveau de la conduite de l'évaluation.

### 2.3.4 Cognitive Walkthrough for the Web

CWW est une méthode destinée à détecter et corriger les erreurs de conception qui peuvent survenir lors de la navigation et la recherche d'informations sur un site Web [Blackmon et al., 2002 ; Kitajima, 2006]. C'est une variante de CW3.

CWW est basée sur une théorie appelée CoLiDeS (Comprehension-based Linked model of Deliberate Search) [Kitajima et al., 2000]. CoLiDeS est en fait une variante du modèle LICAI (Linked model of Comprehension-based Action planning and Instruction taking) qui est un modèle d'exploration basée sur la compréhension [Kitajima et Polson, 1997]. CWW permet de simuler un utilisateur naviguant sur une page avec un but en tête. L'exploration consiste à choisir une action (clic sur une icône, un lien, etc.) puis à évaluer le résultat en fonction du but. CoLiDeS suppose que la sélection d'une action est un processus à deux phases : d'intention, de sélection. Pendant la première, l'utilisateur segmente la page en un ensemble de zones et génère une brève description de chaque zone à partir de sa connaissance de la convention de structuration de page-écran. L'utilisateur s'occupe de la zone dont la description est perçue comme similaire à son but courant. Pendant la phase de sélection de l'action, l'utilisateur génère les descriptions de tous les outils graphiques dans la zone concernée et agit sur celui dont la description est la plus proche de son but. Pour CWW les questions d'évaluation sont les suivantes :

- Q1 : Est-ce que l'action correcte est suffisamment mise en évidence pour l'utilisateur ?
- Q2 : Est-ce que l'utilisateur relie la zone correcte de la page avec le but et ceci en utilisant l'information sur le titre et la compréhension de la convention de structuration de la page ?
- Q3 : Est-ce que l'utilisateur relie le but avec les outils graphiques corrects dans la zone en question et ceci en utilisant des étiquettes des liens et d'autres informations descriptives ?
- Q4 : Est-ce que l'utilisateur interprète correctement la réponse du système avec l'action choisie ?

CWW transforme cette approche en utilisant l'Analyse Sémantique Latente (ASL) [Landauer et Dumas, 1997] : au lieu de faire juger subjectivement par des évaluateurs les réponses à ces questions, on utilise l'ASL permettant d'estimer la liaison sémantique des textes basés sur l'analyse statistique d'un large corpus. Donc ici, l'ASL est utilisée pour estimer la similarité sémantique entre les descriptions des buts de l'utilisateur et les descriptions des zones des pages ainsi que les buts et les descriptions des actions possibles sur la page web.

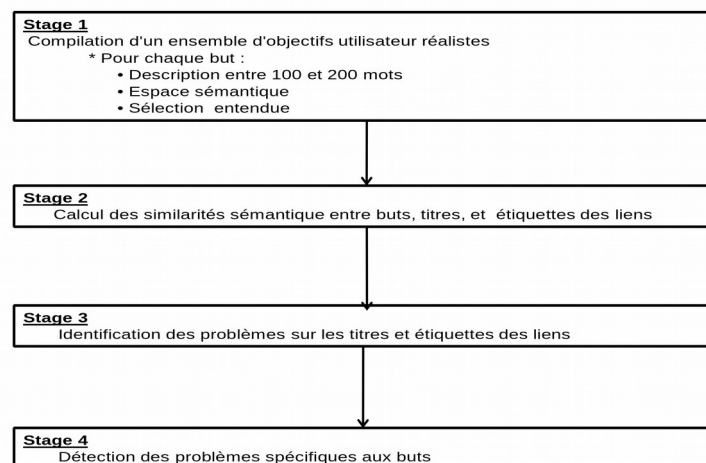


Figure 2.2 Les étapes de CW for the Web (adapté de [Blackmon et al., 2002])

L'évaluation se déroule en 4 phases (figure 2.2). Dans la phase 1 on formule un ensemble de buts utilisateurs réalistes composé de 100 à 200 mots, puis on identifie la sélection correcte prévue pour chaque but. La phase 2 consiste à calculer les similarités sémantiques entre les buts, les titres et les étiquettes des liens. La phase 3 a pour objet l'identification des titres et étiquettes de liens

problématiques en supposant que selon CoLiDeS : un titre cause un problème s'il n'est pas familier, ou s'il est en confusion avec d'autres. Un titre avec une valeur ASL de moins de 0.8 est supposé non familier. Une paire de titres ayant une valeur ASL de plus de 0.6 est supposée confuse. La phase 4 consiste à rechercher les problèmes spécifiques au but.

### 2.3.5 Groupware Walkthrough

Groupware Walkthrough (GW) [Pinelle et Gutwin, 2002], méthode d'inspection pour Groupware, est une modification profonde de CW visant à inclure des facteurs de complexité liés au fonctionnement d'une équipe de travail. GW intègre une modélisation de la tâche et un processus de type walkthrough. Un modèle de tâche permet d'identifier et analyser les tâches collaboratives du monde réel. Le modèle de groupe de tâches doit permettre d'exprimer la variabilité des actions dans le groupe de travail. Un processus Walkthrough vise à évaluer le support du système vis-à-vis des tâches. Les tâches sont modélisées comme un ensemble de chemins possibles pour aboutir à un résultat voulu. Chaque chemin possible est exploré pendant le walkthrough, et le support pour chaque chemin dans l'interface est évalué.

La collaboration implique deux types distincts d'activités : les activité-tâches qui sont les actions qui doivent se produire pour accomplir la tâche, les activités du groupe qui sont les actions que les membres du groupe doivent exécuter pour accomplir ensemble une tâche. Notons que l'approche traditionnelle d'évaluation se focalise sur l'activité tâche. L'activité de groupe peut être analysée suivant une abstraction de haut niveau et une de bas niveau. Le haut niveau incluant les aspects sociaux et organisationnels est traité en s'appuyant sur GTA (Groupware Task Analysis) [Van der Veer et al., 1996 ; Van der Veer et van Velie, 2000]. Dans GTA les méthodologies associées se focalisent sur la spécification des tâches basée sur l'observation ethnographique. GTA est un moyen efficace pour rassembler des informations sur le contexte de travail, mais son niveau d'analyse est trop élevé pour être utilisé dans une situation d'inspection (Walkthrough). Pour capturer le niveau des détails nécessaires pour l'inspection, il faut considérer l'activité du groupe au plus bas niveau c.-à-d. au niveau des processus de collaboration.

Néanmoins l'analyse de haut niveau précède l'analyse de bas niveau. Les processus de collaboration [Gutwin et Greenberg, 2000] sont les activités de base du travail collaboratif, les actions et interactions à petite échelle que les membres du groupe doivent effectuer pour réaliser une tâche de façon collaborative. Elles sont invariables à travers une variété de facteurs sociaux et organisationnels. Les processus de collaboration comprennent la communication explicite, le monitoring, la coordination d'action, la planification, l'assistance et la protection. Les composants principaux d'un modèle de tâches sont les scénarios, les tâches, les sous-tâches (individuelles ou collaboratives), et les actions. Le scénario est une description de haut niveau des activités liées à l'accomplissement d'un résultat spécifique. L'analyse de tâche commence par la collecte des données observées sur le travail, et continue avec l'identification des épisodes d'interaction collaborative. Les scénarios et les tâches peuvent donc être spécifiés et les sous-tâches collaboratives peuvent être analysées selon le processus de collaboration. Pour chaque scénario collaboratif, GW peut ensuite être exécutée de la façon suivante :

- D'abord, se familiariser avec les contextes, l'environnement
- Ensuite pour chaque tâche du scénario :
  - essayer d'exécuter chaque tâche alternative
  - enregistrer comment chaque tâche est exécutée
  - enregistrer les problèmes détectés
- Après l'exécution de chaque tâche, il faut se poser les questions suivantes :



- Est-ce que la tâche peut être exécutée efficacement, est-ce que l'interface fournit le moyen de le faire et correctement ?
- Est-ce que la tâche peut être effectuée de façon efficiente ?
- Est-ce que la tâche est exécutée avec satisfaction ?
- Après l'exécution de toutes les tâches, il faut vérifier si l'interface permet au groupe de réaliser toutes les tâches entendues.

### 2.3.6 Activity Walkthrough

Activity Walkthrough [Bertelsen, 2004] est une version modifiée de CW, basée sur la théorie de l'activité, outil descriptif puissant qui d'une part s'intéresse à la compréhension de l'activité humaine et d'autre part incorpore les notions d'intentionnalité, d'histoire, de médiation, de collaboration et de développement [Nardi, 1996]. Selon la théorie de l'activité, l'unité de l'analyse est l'activité entière. Une activité est composée d'un sujet, d'un objet et des outils qui servent de médiation. Un sujet est une personne ou un groupe de personnes engagées dans une activité. Un objet est tenu par le sujet et motive l'activité. La médiation peut se produire à travers l'utilisation de différents types d'outils : outil matériel, outil mental (y compris la culture), moyens de penser et langage. Selon Kaptelinin [1996], l'ordinateur est considéré comme un outil de médiation. L'activité est réalisée à travers un objectif conscient dirigé par des actions qui à leur tour sont réalisées à travers des opérations inconscientes.

Pour Activity Walkthrough, l'évaluation se fera selon six phases. La phase 1 (de préparation) permet d'identifier les tâches typiques à analyser, en se basant sur la spécification des besoins. La phase 2 (de contextualisation) permet d'identifier les activités dans lesquelles l'application est utilisée comme médiatrice. Pour chaque activité, il faut identifier d'une part les actions à travers l'application qui contribuent à la réalisation de l'activité et d'autre part les objets et résultats de ces actions. Il faut considérer aussi les autres moyens de réalisation de l'activité sans l'application, c'est-à-dire d'autres artefacts utilisés comme médiatrice de l'activité. Il faut considérer aussi l'horizon d'attente de l'utilisateur, par ex. son expérience en utilisant des applications ou outils similaires. La phase 3 (de vérification des tâches) consiste à évaluer sur la base d'une contextualisation de l'application, dans quelle mesure chaque tâche correspond aux actions déterminées dans les activités dans lesquelles l'application sera implantée. La phase 4 (analyse des tâches) consiste à décomposer chaque tâche en séquence d'opérations atomiques comme dans CW. La phase 5 (phase d'évaluation) consiste à répondre à trois questions relatives au contexte perceptuel, aux réponses du système et à l'apprentissage d'utilisation du système. La phase 6 (de vérification) consiste à analyser la tâche en se basant sur la phase précédente. Suite à ces six phases, un rapport de synthèse doit être établi pour conclure l'évaluation.

### 2.3.7 Interaction Walkthrough

La méthode Interaction Walkthrough [Ryu et Monk, 2004] est une version modifiée de CW, basée sur une théorie dite de l'interaction cyclique (figure 2.3), dont l'idée a initialement été introduite par Card et al. [1983] dans leur cycle de reconnaissance-action. Le modèle en 7 étapes de Norman [1986] imagine aussi un cycle d'interaction ; néanmoins, aucun de ces auteurs n'énonce formellement comment l'environnement et les contextes influencent l'interaction de l'utilisateur. Le moyen à la disposition de l'utilisateur pour manipuler le système est centré sur l'action (« chemin but-action »). Une action avec le dispositif d'entrée déclenche des effets système (« chemin action-effets »). Dès la fin de la phase d'exécution, celle d'évaluation commence, c'est ainsi que le système rentre dans un nouvel état. Le « chemin effets-buts » s'occupe des changements sur ceux qui sont perçus et continue sur des nouveaux buts dans le contexte d'interaction. Les trois chemins

d'interaction donnent ainsi trois classes de problèmes de bas niveau : de but-action, d'action-effet, d'effet-but.

Les problèmes de « but-action » sont souvent observés dans le cas où des actions imprévisibles sont conçues pour accomplir un objectif. Une action imprévisible peut être due à une faible affordance de l'action correcte ou à une forte affordance de l'action incorrecte. Selon Norman [1988, 1999], l'affordance réfère aux attributs des objets qui permettent à l'homme de savoir comment les utiliser.

Les problèmes d'« action-effet » concernent les problèmes de mode, dans lequel une même action produit différents effets. On distingue trois types de problème de mode. Le premier concerne les problèmes de mode caché. Dans ce cas les erreurs ont tendance à se produire quand le changement de mode a été oublié. Le deuxième concerne les problèmes de mode partiellement caché. Dans ce cas, les erreurs ont tendance à se produire quand on est incapable de reconnaître l'environnement contenant l'information qui indique le mode. Et enfin le troisième concerne les problèmes de signal de mode erroné. Dans ce dernier cas, les erreurs ont tendance à se produire quand un signal de mode est perçu par l'utilisateur alors qu'il ne devrait pas l'être, ce qui trompe les utilisateurs qui croient qu'ils exécutent l'action correcte.

Le chemin « effet-but » peut être considéré comme un processus de réorganisation de but. Ce qui entraîne deux types de problèmes : les problèmes de construction de but et les problèmes d'élimination de but. Les problèmes de construction de but se produisent quand le but est ambigu ou hors de propos. Tandis que les problèmes d'élimination de but se produisent suite à l'absence ou la présence trompeuse de réactions du système.

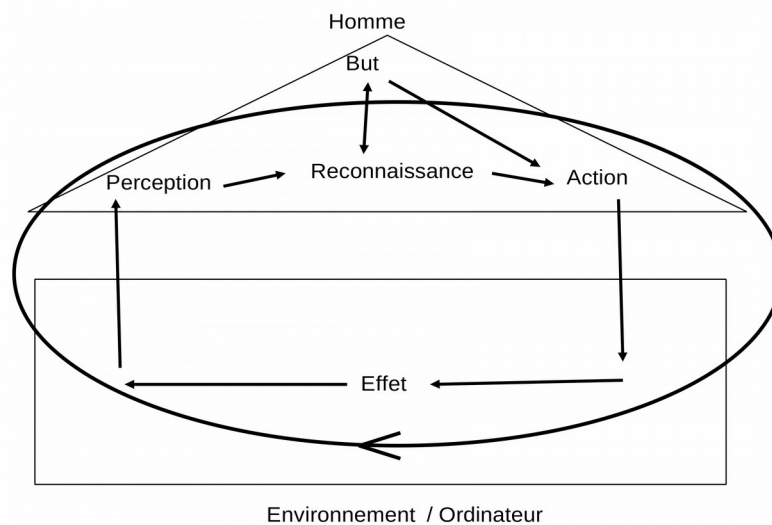


Figure 2.3 Théorie de l'interaction cyclique [Monk, 1998]

L'évaluation concerne les trois chemins du cycle d'interaction que nous venons de décrire et se déroule en 3 phases pour chaque type de problème : d'affordance, de construction et d'élimination de but, de mode. D'abord la phase de préparation consiste à localiser les problèmes. Ensuite la phase Walkthrough permet de détecter les problèmes en répondant à des questions. Enfin la phase de vérification consiste à réexaminer les problèmes trouvés précédemment pour vérifier s'ils suivent les chemins suivis par le concepteur. Les détails des 4 procédures d'évaluation des 4 types de problèmes cités plus haut sont disponibles dans [Ryu et Monk, 2004].

### 2.3.8 Cognitive Walkthrough with users

CW with users [Granollers et Lorés, 2005] est une version incluant explicitement les utilisateurs au cours du processus. Elle se déroule en 3 étapes. La première phase consiste à effectuer CW de manière traditionnelle. La deuxième consiste à incorporer les utilisateurs de la façon suivante : il faut d’abord recruter des utilisateurs qui sont représentatifs des profils des futurs utilisateurs finaux. Après une explication introductive, chaque utilisateur est invité à accomplir l’ensemble des tâches définies dans le Walkthrough correspondant à leur profil. Ils sont invités à exprimer à haute voix leurs pensées, sensations et opinions sur n’importe quel aspect du système ou du prototype pendant qu’ils interagissent avec. Chaque utilisateur accomplit les tâches sans aucune explication complémentaire et à la fin de chaque tâche il doit noter les principaux défauts et imperfections détectés. De plus, une fois que l’utilisateur a terminé les tâches, il est invité à commenter les problèmes identifiés pour cerner son point de vue. Enfin la troisième phase invite les experts à revoir *a posteriori* les doutes ressentis lors à la deuxième phase.

### 2.3.9 Extended Cognitive Walkthrough

L’utilisation de CW nécessite souvent des connaissances en psychologie cognitive. Kato et Hori [2005] constatent que certains évaluateurs éprouvent des difficultés à comprendre les différences entre effet correct et action correcte, et aussi entre différentes questions. Kato et Hori [2005 ; 2006] tentent de contourner ces problèmes. Ils proposent pour cela une variante de la 3ème version de CW ; elle est appelée Extended Cognitive Walkthrough (Extended CW).

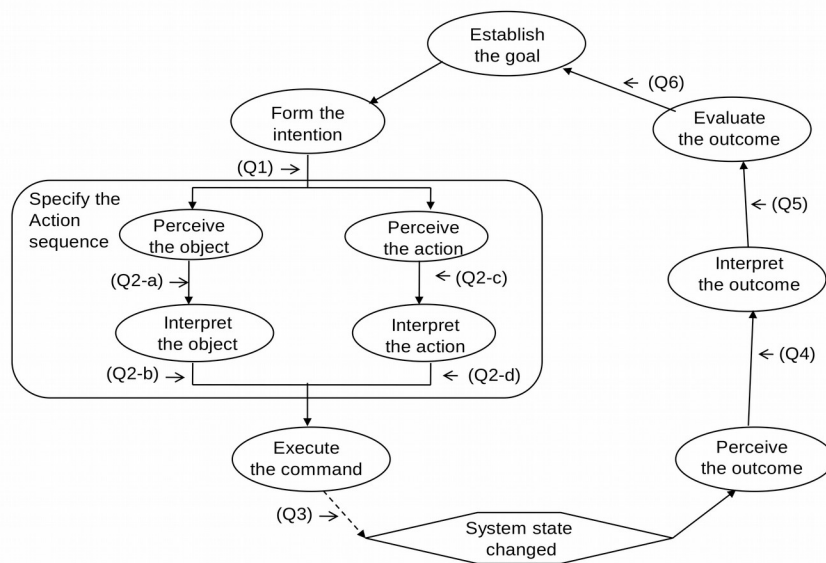


Figure 2.4 : Modèle de Norman étendu adapté de [Kato et Hori, 2005]

Le modèle théorique de la méthode est le modèle de l’action de Norman [Norman, 1986] étendu selon la figure 2.4. Dans celui-ci, la modification concerne la troisième étape en l’éclatant pour qu’on puisse percevoir et interpréter séparément l’objet et l’action. Pendant la phase d’évaluation, on doit répondre aux 9 questions suivantes au lieu de 4 pour CW, se rapportant directement au modèle de Norman étendu (figure. 2.4) :

- (Q1) Will the user intend to achieve the right effect?
- (Q2-a) Will the user notice that the correct object is available?
- (Q2-b) Will the user know what the correct object refers to?
- (Q2-c) Will the user notice that the correct action is available?
- (Q2-d) Will the user know that the correct action should be applied to the correct object?
- (Q3) Will the user be able to apply the correct action to the correct object without fail or difficulty?
- (Q4) When the correct action is taken, will the user notice the physical change in the system state?
- (Q5) Will the user know what exactly has happened to the system state?
- (Q6) Will the user know that the current system state is nearer to completion of the task?

Une information est accessible si elle est facilement perceptible par les utilisateurs. Selon les auteurs de la méthode, l'évaluation de l'accessibilité d'une information doit être faite au niveau de sa perception et de sa compréhension. Cette méthode est donc adéquate pour la détection des problèmes d'accessibilité (questions Q2-a et Q2-c pour la perception et Q2-b et Q2-d pour la compréhension). Les auteurs de la méthode constatent que Extended CW possède une plus large possibilité d'identifier des problèmes d'accessibilité que la troisième version de CW.

### **2.3.10 Distributed Cognitive Walkthrough**

Distributed Cognitive Walkthrough (DCW) est une variante de CW3 [Eden, 2007, 2008]. Elle est basée sur deux théories : celle de la cognition distribuée [Hollan et al., 2000] et celle de la tâche cognitive distribuée [Zhang et Norman, 1994].

Selon Hollan et al. [2000], la théorie de la cognition distribuée, comme toute autre théorie cognitive, cherche à comprendre l'organisation des systèmes cognitifs. Par contre, elle étend ce qu'on attend par cognitif au-delà de l'individuel pour inclure les interactions entre l'homme et les objets de l'environnement.

L'application de la théorie de la cognition distribuée à l'observation de l'activité humaine implique trois types de distributions de processus : (1) les processus cognitifs peuvent être distribués à travers les membres du groupe social ; (2) les processus cognitifs peuvent invoquer une coordination entre la structure interne et la structure externe (Voir principe de la distribution ci-après) ; (3) les processus cognitifs peuvent être distribués à travers le temps de telle sorte que les produits des événements précédents puissent transformer la nature des événements suivants.

La représentation des informations influence le comportement cognitif d'un individu pour une même structure formelle de tâche. La théorie des tâches cognitives distribuées apporte sa contribution aux problèmes de la représentation distribuée des informations, l'interaction entre la représentation interne et la représentation externe, et la nature des représentations externes.

Le principe de la distribution est qu'une partie de la tâche est représentée en interne (représentation mentale) et que l'autre partie est représentée en externe (dans l'environnement physique) [Zhang et Norman, 1994]. La représentation interne peut concerner des propositions, des images mentales, des règles de production, etc. ; tandis que la représentation externe peut concerner les symboles physiques, des règles et contraintes externes, des relations incrustées dans des configurations physiques, etc.

Comme CW, DCW utilise 4 questions auxquelles l'évaluateur doit répondre, de telle sorte qu'une réponse négative invoque de problèmes d'utilisabilité :

- Q1: Will the way that information is represented provide all the knowledge required to carry out the task?
- Q2: Will the way that information is represented show relevant previous progress towards completing the overall task?
- Q3: Will the way that information is represented provide resources that keep the user from having to figure out or calculate anything in his/her head while carrying out the task?
- Q4: If the current task is accomplished, will the way that information is represented be changed in any way so that the result of the task is accessible to current or future users at a later time or in a different place?

### 2.3.11 Enhanced Cognitive Walkthrough

Enhanced Cognitive Walkthrough (Enhanced CW) est une variante de CW3. Elle a été proposée par [Bligard et Osvalder, 2007] et est utilisée dans une approche analytique de prédiction et d'identification de problèmes d'utilisabilité et d'erreurs d'utilisation. L'erreur d'utilisation est selon l'IEC [2004] « *an act or omission of an act that has a different result than intended by manufacturer or expected by the operator* ».

L'approche proposée par [Bligard et Osvalder, 2007] comporte quatre phases : (1) la définition de l'évaluation, (2) la description du système, (3) l'analyse de l'interaction et (4) la production et la rédaction des résultats. Enhanced CW intervient dans la phase d'analyse de l'interaction en parallèle avec une analyse prédictive d'erreurs d'utilisation [Bligard, 2007].

Enhanced CW se déroule en deux étapes. La première vise la prédiction des problèmes d'utilisabilité ; elle consiste à inspecter les séquences d'actions en utilisant les questions suivantes :

- Q1: Will the user know that the evaluated function is available?
- Q2: Will the user try to achieve the right effect?
- Q3: Will the user interface give clues that show that the function is available?
- Q4: Will the user be able to notice that the correct action is available?
- Q5: Will the user associate the right clue with the desired function?
- Q6: Will the user associate the correct action with the desired effect?
- Q7: Will the user get sufficient feedback to understand that the desired function has been chosen?
- Q8: If the correct action is performed, will the user see that progress being made towards the solution of the task?
- Q9: Will the user get sufficient feedback to understand that the desired function has been performed?

La réponse à chaque question est un nombre compris entre 1 et 5. 1 signifie Non, 2 signifie probablement Non, 3 signifie que la réponse est inconnue, 4 signifie probablement Oui et enfin 5 signifie Oui. Comme la troisième version de CW, chaque réponse doit être justifiée. Toute réponse à une question différente de 5 suggère l'existence probable d'un problème d'utilisabilité.

La deuxième étape de Enhanced CW est l'identification des problèmes ; elle consiste à déterminer la catégorie de chaque problème. La catégorisation est basée sur la justification et la description du problème. On peut avoir plusieurs catégories de problème selon le type de l'IHM et la tâche concernés. Bligard et Osvalder [2007] proposent cinq catégories de problèmes d'utilisabilité :

- Les problèmes liés à l'expérience et la connaissance de l'utilisateur ;
- Les problèmes liés à l'affichage de l'information, c'est-à-dire que l'interface ne donne aucune indication sur la fonction disponible ou sur la façon de l'utiliser ;
- Les problèmes liés à l'illustration de l'information, c'est-à-dire que le placement, l'apparence et le contenu de l'interface peuvent être mal interprétés ou non compris ;
- Les problèmes liés à la séquence d'actions, c'est-à-dire que les actions sont peut être exécutées de façon non naturelle ;
- Les problèmes liés au retour d'information, c'est-à-dire que l'interface ne donne pas assez d'indications sur ce que fait ou a fait l'utilisateur.

## 2.4 Synthèses

Nous proposons quatre synthèses complémentaires qui sont successivement décrites. La première aborde les aspects conceptuels, méthodologiques et technologiques de chaque version ou variante. La seconde présente des études comparatives entre au moins une version ou une variante de CW et au moins une autre d'évaluation. La troisième s'intéresse à des études non comparatives (c'est-à-dire que les méthodes sont liées uniquement à CW). La quatrième synthèse est une tentative destinée à faciliter le choix entre les versions et variantes de CW.

### 2.4.1 Synthèses selon les aspects conceptuels, méthodologiques et technologiques

Au début, CW était une méthode destinée à évaluer les problèmes d'utilisabilité des systèmes de type « Walk-up-and-use », c'est-à-dire qu'on utilise sans ou avec peu de formation initiale. Son succès est tel que les professionnels et les chercheurs s'intéressent à cette méthode pour l'améliorer davantage ou pour l'adapter à d'autres types d'application plus ou moins spécifiques. Onze versions modifiées ou étendues de CW ont été présentées. Les modifications et les extensions effectuées concernent dans la plupart des cas les aspects conceptuels ou théoriques, d'autres concernent en plus les aspects méthodologiques et deux versions seulement concernent en partie les aspects technologiques.

La première version de Lewis et al. [1990] nécessite implicitement une connaissance de base en science cognitive pour bien comprendre la terminologie utilisée dans les formulaires. La compréhension de la distinction entre but et action est une condition cruciale pour la réussite de la méthode de la part des évaluateurs. Pour faire face à ce problème, Polson et al. [1992] proposent la seconde version plus formelle qui entraîne les évaluateurs à faire des analyses détaillées pour éviter les ambiguïtés. Le modèle d'action de Norman et le modèle de construction-intégration de Kintsch permettent à cette méthode de faire des analyses détaillées et structurées selon les buts et les actions concernés. Mais avec le nombre important de formulaires à remplir et les listes de questions auxquelles il faut répondre, la méthode devient plus complexe et plus lourde à appliquer. Des outils ont été développés [Rieman et al., 1991] pour aider les évaluateurs à remplir les différents formulaires et à se concentrer sur l'évaluation. Néanmoins, même avec cette aide précieuse, les

auteurs de la méthode ont constaté que celle-ci est coûteuse en temps et en énergie dépensée pour son application. La 3ème version tente de réduire la complexité et les coûts d'exploitation en introduisant des modifications sur l'aspect méthodologique. Néanmoins la méthode nous semble loin d'être parfaite dans sa troisième version. De manière plus générale, notons aussi que la plupart des auteurs des différentes variantes citées ont émis des remarques, critiques et suggestions en lien avec les points faisant l'objet de leur variante.

En ce qui concerne Heuristic Walkthrough, les modifications ont été apportées au niveau méthodologique. En effet après la première phase qui suit un processus CW entier, la seconde phase pourrait être considérée comme une redondance en termes d'évaluation, c'est-à-dire qu'on évalue une seconde fois le système ce qui devient assez lourd. Mais selon l'expérimentation de Sears [Sears, 1997], cette redondance permet à Heuristic Walkthrough de trouver plus de problèmes d'utilisabilité que dans CW. C'est un résultat prometteur qui appelle à des études complémentaires.

Concernant Norman's CW, le modèle de l'action est adopté comme fondement théorique, ce qui permet de traiter le problème à un haut niveau d'abstraction et de localiser l'origine des problèmes lors de l'évaluation. Au niveau méthodologique, les questions sont focalisées à l'échelle des distances cognitives facilitant ainsi la localisation des problèmes que nous venons de citer. Dans le document de référence sur ces travaux [Rizzo et al., 1997] les auteurs ne fournissent aucune information pour juger la validité de cette méthode.

Les modifications incarnées par Streamlined CW se situent au niveau des aspects méthodologiques et consistent d'abord à bien préparer l'équipe d'évaluation, ensuite de bien encadrer le processus en imposant des règles de base, et enfin à reformuler les quatre questions initiales en deux questions. Ces modifications permettent de maîtriser la contrainte du temps, puis d'éliminer les longues discussions concernant la conception pendant l'évaluation et enfin d'empêcher les concepteurs de défendre à tort leurs points de vue très influencés par leur expérience de conception. Il nous semble que le fait de généraliser les questionnaires au lieu de les détailler peut compromettre l'efficacité de la méthode pour des évaluateurs novices. C'est un point qui pourrait être étudié lors d'expérimentations futures.

Sur le plan de l'aspect conceptuel, CW for the Web adopte le modèle cognitif CoLiDeS. De plus c'est la seule variante qui incarne une modification sur les aspects technologiques. En effet c'est une variante destinée spécifiquement à évaluer les problèmes d'utilisabilité des sites Web, les quatre questions sont focalisées sur la navigation et la recherche d'information. Par ailleurs, pour contourner les jugements subjectifs concernant les réponses aux quatre questions, une analyse sémantique latente est utilisée. Pour cela on utilise une description riche des buts permettant d'incorporer plus d'informations concernant la compréhension des utilisateurs de leurs tâches et arriver ainsi à obtenir des buts réalistes. Le choix des espaces de connaissances sémantiques appliqué à l'Analyse Sémantique Latente permet à CWW de tenir compte de la population cible (par ex. selon l'âge) lors de l'évaluation. Il nous semble que l'efficacité de cette méthode dépendra donc de la qualité de l'espace de connaissances sémantiques choisi et de son interprétation.

Groupware Walkthrough est la seule variante destinée à évaluer des applications de type CSCW. Elle est basée sur un modèle de processus de collaboration pour faire face à la complexité et aux spécificités induites par ces applications, entre autres les problèmes d'utilisabilité liés à la collaboration. Elle permet aussi d'étudier en détail les contextes de chaque tâche se situant dans le cadre de l'évaluation grâce à GTA [Van der Veer et al. 1996 ; Van der Veer et van Velie, 2000]. Néanmoins, elle ne permet pas de détecter des problèmes d'utilisabilité classiques. Il semble également qu'elle nécessiterait des améliorations pour permettre de détecter et prendre en compte

les différents aspects socio-organisationnels d'utilisation dans un contexte réel d'un système collaboratif [Pinelle et Gutwin, 2002].

Activity Walkthrough est basée sur la théorie de l'activité, ce qui permet aux évaluateurs utilisant cette méthode d'inclure plus de contexte dans l'évaluation et l'engagement dynamique dans l'action, puisque l'élément de base de l'analyse est l'activité réelle des utilisateurs. Cependant, il nous semble que si on aborde CW par la théorie de l'activité qui exige la prise en compte de nombreux facteurs notamment liés à la variabilité humaine intra et inter individuelle, mais aussi aux contextes d'utilisation, alors on complique indéniablement l'évaluation en terme de lourdeur et de difficulté d'interprétation (ce qui exige aussi une grande expertise en terme d'évaluation). Ce point va d'ailleurs dans le sens de l'auteur de la méthode [Bertelsen, 2004] ainsi que de celui d'autres auteurs tel [Ryu, 2008] qui insistent sur la complexité et la lourdeur de la méthode.

Interaction Walkthrough est basée sur le modèle dit de l'interaction cyclique. Elle considère les problèmes plutôt au bas niveau. Par opposition aux autres méthodes, qui sont censées détecter les problèmes plus ou moins compliqués en les localisant et en expliquant leurs origines, Interaction Walkthrough détecte plus facilement les problèmes puisqu'ils sont de bas niveau, c'est-à-dire qu'ils sont déjà explicités par la méthode. Cependant, si on se réfère à [Ryu et Monk, 2004] qui pensent que le comportement de l'utilisateur est dépendant de la technologie utilisée et que cette technologie doit être vue à l'intérieur d'une situation d'activité ayant un sens pour l'utilisateur et dans un large contexte, on peut douter de la validité des résultats de cette méthode vue la complexité liée à l'analyse de l'interaction.

CW With Users correspond à une méthode CW s'élargissant à une vraie intervention des utilisateurs dans le sens plus large du terme (celle en particulier de l'ergonomie francophone). C'est une variante intéressante, sous l'angle de l'ergonomie, mais n'a-t-elle pas pour conséquence de faire sortir la méthode de la catégorie des méthodes d'inspection destinée à des experts uniquement, puisqu'elle fait intervenir des experts et des utilisateurs ? Ceci nous amène à nous poser une autre question fondamentale : les modèles et méthodes d'intervention et d'analyse choisis sont-ils les plus adaptés (par exemple utilisation de la verbalisation simultanée à haute voix qui pourrait perturber le raisonnement et la pensée naturelle de l'utilisateur) ?

Du point de vue conceptuelle, Extended CW est une modification de CW adoptant un modèle étendu de la théorie de l'action de Norman ; ce qui permet d'évaluer l'accessibilité de l'information en tenant compte de tous les aspects cognitifs dès la perception, jusqu'à l'exécution de l'action tout en passant par la compréhension. Sur l'aspect méthodologique, les questions sont focalisées sur les distances cognitives comme dans Norman's CW. Kato et Hori [2006] concluent après avoir expérimenté la méthode que Extended CW détecte plus de problèmes d'utilisabilité que CW dans un temps comparable. Bien que les fondements théoriques de la méthode nous semblent solides, la spécificité des conditions expérimentales (type des participants, rémunération, types d'IHM évaluées) décrites dans [Kato et Hori, 2005] ne permet pas de conclure sur une généralisation quant à l'efficacité de cette variante dans d'autres conditions d'utilisation (selon les critères de validité proposées par Gray et Salzman [1998]).

Distributed Cognitive Walkthrough est une variante de CW3 adoptant les théories de la cognition distribuée. Donc théoriquement, cette méthode permet d'identifier des problèmes d'utilisabilité à travers des dimensions telles que le temps, l'espace et la structure sociale. En revanche, l'auteur de la méthode n'a pas fourni de résultats d'expériences d'utilisation de la méthode.

Enfin, Bligard et Osvalder [2007] soulignent l'efficacité de Enhanced CW dans le cadre de l'approche analytique de prédiction et d'identification des problèmes d'utilisabilité et des erreurs



d'utilisation. Il nous semble qu'il reste à démontrer son efficacité si on l'utilise seule en dehors de ce cadre.

Le tableau 2.1 propose une synthèse de tous les versions et variantes citées dans ce chapitre. Trois colonnes différencient les aspects théoriques et conceptuels, méthodologiques et technologiques.

Pour terminer cette synthèse nous tenons à insister sur le fait que les variantes de CW, bien qu'elles ne soient pas homogènes, simulent ce qui se produit dans les pensées des utilisateurs en exécutant leurs tâches avec un système particulier (aspect conceptuel) ; il s'agit d'évaluer ce système en répondant à des questions spécifiques (aspect méthodologique) et en utilisant éventuellement différents outils et techniques (aspect technologique). Faute de place nous ne pouvons pas détailler ici les questions posées par chaque méthode mais en général, elles ont pour but d'évaluer à chaque interaction si les utilisateurs arrivent à i) formuler de buts, ii) suivre et exécuter les actions appropriées pour atteindre les buts, iii) interpréter correctement les réponses du système, iv) pour finalement atteindre correctement les buts.

Tableau 2.1 : Synthèse de toutes les versions et variantes de CW

Nom	Aspects théoriques et conceptuels	Aspects méthodologiques	Aspect technologiques
<b>CW1</b> [Lewis <i>et al.</i> , 1990]	Modèle de l'apprentissage par exploration CE+	1. Préparation : choix de tâches à évaluer, décomposition des tâches en suite d'actions. 2. Evaluation avec un formulaire.	- sans outil matériel proposé.
<b>CW2</b> [Polson <i>et al.</i> , 1992]	Modèle de l'apprentissage par exploration CE+, Modèle de l'action de Norman, Modèle la construction-intégration	1. Préparation : choix de tâches à évaluer, décomposition des tâches en suite d'actions, identification des utilisateurs cible, description de la structure de buts initiale. 2. Evaluation avec un formulaire.	- Formulaire d'évaluation : structure de buts, choix et exécution de l'action, modification de la structure de buts. - Proposition d'une version automatisée (Rieman <i>et al.</i> , 1991)
<b>CW3</b> [Wharton <i>et al.</i> , 1994]	Modèle de l'apprentissage par exploration CE+	1. Préparation : choix de tâches à évaluer, description en détail de chaque scénario. 2. Evaluation prenant en compte des détails des scénario.	- Proposition de guide
<b>Heuristic Walkthrough</b> [Sears, 1997]	Modèle de l'apprentissage par exploration CE+	Combinaison de CW, l'évaluation heuristique.	- Liste des heuristiques.
<b>Norman' CW</b> [Rizzo <i>et al.</i> , 1997]	Modèle de l'action de Norman	Identique à CW3 mais avec des questionnaires se rapportant au modèle de Norman.	- Questionnaires focalisés sur les actions.
<b>Streamlined Cognitive Walkthrough</b> [Spencer, 2000]	Identique à CW3	Identique à CW3 avec renforcement du contrôle de l'évaluation	- Proposition des questionnaires - Proposition des règles de conduit de l'évaluation.
<b>Cognitive Walkthrough for the Web</b> [Blackmon <i>et al.</i> , 2002; Kitajima, 2006]	Modèle CoLiDeS	4 étapes : constitution d'un ensemble de buts réalistes et d'une sélection entendue, évaluation des similarités sémantiques avec LSA, identification des problèmes sur les titres et les étiquettes des liens, détection des problèmes spécifiques aux buts.	- Outils d'analyse disponible à: <a href="http://autocww.colorado.edu/HomePage.html">http://autocww.colorado.edu/HomePage.html</a>
<b>Groupware Walkthrough</b> [Pinelle et Gutwin, 2002]	Dédié au Groupware (processus collaboratif)	- Modélisation des tâches (scénario, tâche, sous-tâche) - Analyse des tâches avec GTA - Evaluation de scénario	- Proposition des questionnaires

Nom	Aspects théoriques et conceptuels	Aspects méthodologiques	Aspect technologiques
<b>Activity Walkthrough</b> [Bertelsen, 2004]	Théorie de l'activité	7 étapes : identification de tâche typique à analyser, contextualisation, vérification de tâche, analyse de tâche, évaluation, vérification globale, production de rapport	- Proposition des questionnaires - Rapport de synthèse
<b>Interaction Walkthrough</b> [Ryu et Monk, 2004]	Théorie de l'interaction cyclique (Monk, 1998)	- Détection des problèmes d'utilisabilité se rapportant à l'interaction (but-action, action-effet, construction de but et élimination de but) - 4 évaluations composées chacune de 3 étapes (préparation, évaluation, vérification)	- Proposition des questionnaires
<b>Cognitive Walkthrough with User</b> [Granoller et Lorés, 2005]	- CW impliquant les utilisateurs. - Verbalisation de l'utilisateur	- CW avec intervention des utilisateurs réels. - Se déroule en deux étapes: CW avec expert, intervention des utilisateurs	- Proposition des questionnaires
<b>Extended Cognitive Walkthrough</b> [Kato et Hori, 2005]	Modèle de Norman étendu	Identique à CW3 avec des questionnaires focalisant sur les distance sémantiques identifiées par le modèle de Norman étendu.	- Pas d'outil supplémentaire proposé
<b>Distributed Cognitive Walkthrough</b> [Eden, 2007]	- Théorie de la cognition distribuée - Théorie de la tâche cognitive distribuée	Identique à CW3 avec des questionnaires focalisés sur la cognition distribuée	- Pas d'outil supplémentaire proposé
<b>Enhanced Cognitive Walkthrough</b> [Bligard et Osvalder, 2007]	Modèle de l'apprentissage par exploration CE+	Considéré comme une partie d'une approche analytique de prédiction et d'identification d'erreurs d'utilisation et de problèmes d'utilisabilité.	- Proposition des questionnaires comme CW3 et d'une liste de types de problème

## 2.4.2 Synthèse sur les études comparatives

Nous résumons dans cette section certaines études dans lesquelles sont évaluées les performances de différentes versions et variantes de la méthode CW.

Le tableau 2.2 récapitule l'ensemble de ces études comparatives. Dans celles-ci, une des trois versions de CW est comparée avec une version ou variante de CW, ou au moins un autre moyen d'évaluation. Le système évalué est précisé, de même que les types d'évaluateurs ayant participé à l'étude. En outre, les principaux résultats chiffrés concernant la détection des problèmes d'utilisabilité sont fournis. Pour préciser les différents éléments du tableau, on résumera ci-dessous pour chaque étude, les conditions de l'étude et les conclusions correspondantes.

### 2.4.2.1 Etude de Lewis et al. [1990]

#### Conditions de l'étude :

Ces auteurs ont évalué deux tâches pour chacune des quatre conceptions différentes de l'interface utilisateur d'un système de messagerie. Pour cela, quatre évaluateurs, dont trois familiers de la théorie CE+, ont participé à l'évaluation. Chaque évaluateur a évalué indépendamment les deux tâches pour chacune des quatre interfaces. Les résultats ont été comparés avec les données provenant d'une évaluation empirique précédemment disponible ; ces données sont issues des tests effectués sur les mêmes tâches réalisées sur les quatre interfaces, par au moins 15 sujets.

#### Conclusions de l'étude :

Les auteurs mettent en évidence vingt problèmes dont 18 ont été trouvés par au moins deux des quatre évaluateurs. Ensuite ils ont comparé les résultats des évaluations avec CW1, avec les données empiriques. Cette comparaison leur a permis de conclure que l'exploitation de CW1 a permis de détecter 50% des problèmes d'utilisabilité détectés à l'aide de la méthode empirique.

### 2.4.2.2 Etude de Polson et al. [1992]

#### Conditions de l'étude :

Polson et al. [1992] rapportent les résultats des tests de CW2 sur des projets d'étudiants. Trois interfaces ont été évaluées. Il s'agit d'un système de messagerie vocal (voice mail directory), d'un système de traitement de texte (text editor) et d'un "document routing". Le premier système a été testé, d'une part avec CW2, et d'autre part avec un test utilisateur à haute voix. Le deuxième système a été testé avec CW2 sur une cinquantaine de tâches variées, puis les résultats ont été comparés à ceux obtenus par une méthode empirique (rapports d'erreurs et interviews utilisateurs). Enfin le dernier système a été évalué avec CW2 sur une tâche assez simple, les résultats ont été comparés avec ceux issus de tests d'utilisabilité ultérieurs.

#### Conclusions de l'étude :

Le test sur la première interface a montré que la méthode CW2 a permis de détecter 50% des problèmes d'utilisabilité trouvés par le test utilisateur à haute voix. En ce qui concerne la deuxième interface, les données de comparaison ne coïncident pas spécifiquement avec les tâches utilisées lors des tests sur CW2, néanmoins 30% des problèmes d'utilisabilité ont été détectés en utilisant CW2 (dont 70% des problèmes jugés sérieux). Enfin pour la troisième interface, tous les problèmes d'utilisabilité détectés par les tests d'utilisabilité ultérieurs, l'ont été aussi en utilisant CW2.

### 2.4.2.3 Etude de Sears [1997]

#### Conditions de l'étude :

Cette étude avait pour objectif de comparer les performances obtenues en utilisant les méthodes CW3, Heuristic Walkthrough (HW) et Heuristic Evaluation (HE). vingt étudiants en informatique qui avaient reçu des cours de conception et d'évaluation d'IHM ont participé à l'évaluation d'un système d'apprentissage des effets visuels d'algorithmes de restitution. Les évaluations ont été effectuées par des groupes d'évaluateurs composés de un à cinq participants.

#### Conclusions de l'étude :

Les résultats de l'étude ont montré que HW a permis d'identifier significativement plus de problèmes de sévérité mineure et intermédiaire que CW3 pour les groupes de deux à cinq participants. Tandis que HE a permis d'identifier significativement plus de problèmes de sévérité mineure que CW3 pour les groupes de deux à cinq participants (cf. tableau 2.2). En revanche, HW et CW3 ont permis d'identifier significativement moins de faux problèmes que HE pour les groupes de deux à cinq participants.

### 2.4.2.4 Etude de Granoller et Lorès [2004]

#### Conditions de l'étude :

CWU a été expérimenté sur un simulateur de microscope et un système de gestion de conférence. Pour le premier système, un évaluateur expert a évalué l'application avec CW3. Puis dix étudiants en médecine ont effectué individuellement la deuxième phase de CWU. Pour le deuxième système, un évaluateur expert a évalué l'application. Ensuite trois utilisateurs de profils différents représentant les utilisateurs du système de gestion de conférence ont effectué indépendamment la deuxième phase de CWU.

#### Conclusions de l'étude :

Les résultats ont montré qu'avec CW3, l'évaluateur expert a trouvé cinq problèmes d'utilisabilité sur le premier système. Et en utilisant CWU, on a trouvé onze problèmes d'utilisabilité en plus. Pour le deuxième système, l'évaluateur expert a trouvé neuf problèmes dans la première phase. Dans la seconde les trois utilisateurs ont trouvé 14 problèmes d'utilisabilité en plus.

### 2.4.2.5 Etude de Kato et Hori [2005]

#### Conditions de l'étude :

Cette étude avait pour objectif de comparer CW3 et Extended CW. Pour cela, vingt étudiants ayant une expérience de CW3 ont participé à l'étude. Deux systèmes interactifs (une interface d'un appareil photo numérique et un logiciel de dessin) ont été évalués. On a demandé aux participants d'évaluer sur une échelle de sept points la facilité de compréhension des questions ainsi que la facilité de les répondre pour CW3 d'une part, et de Extended CW d'autre part, puis de comparer les deux versions. Ensuite, il leur était demandé de préciser quelle version ils souhaiteraient utiliser dans le futur.

#### Conclusions de l'étude :

Les problèmes potentiels d'utilisabilité trouvés sur l'interface de l'appareil photo numérique en utilisant CW3 étaient au nombre de 43 contre 59 avec Extended CW. En revanche, sur le logiciel

de dessin, CW3 a permis de détecter 70 problèmes d'utilisabilité contre 62 pour Extended CW. Les résultats ont montré aussi que les questions de Extended CW étaient plus faciles à comprendre que celles de CW3. En outre, 70% des participants ont jugé qu'il était plus facile de répondre aux questions de Extended CW qu'à celles de CW3. C'est pourquoi, 70% d'entre eux ont émis une préférence pour Extended CW en termes d'utilisation dans le futur.

#### **2.4.2.6 L'étude d'Eden [2008]**

##### **Conditions de l'étude :**

L'objectif de cette expérimentation était de comparer DCW et CW3. La comparaison concerne plusieurs critères dont « average severity rating », « usability relevance rating » ainsi que « actionability rating ». Le premier (« average severity rating ») concerne toutes les questions de la méthode utilisée lors d'une session d'évaluation complète ; selon l'auteur, « this measure of the average severity rating provides the most direct assessment of the usability of a design for a task, with respect to the evaluation method used ». Le second critère (« usability relevance rating ») fournit une estimation globale concernant à quel point chaque évaluation complète conduit à des résultats pertinents par rapport à l'utilisabilité. Ce critère permet donc de mesurer la pertinence du résultat de l'évaluation effectuée par un évaluateur quelconque en utilisant la méthode en question. Le troisième (« actionability rating ») exprime dans quelle mesure les descriptions des problèmes d'utilisabilité indiquent le plus explicitement possible ce qu'il faut changer pour remédier aux problèmes.

Un scénario tiré du domaine de la restauration a été étudié. Pour ce scénario, les différentes commandes de boissons doivent être prises par des serveurs ; et ces commandes doivent être suivies jusqu'à ce que les boissons soient préparées et servies aux clients. Les participants sont des étudiants en licence d'informatique qui n'ont pas d'expérience sur les méthodes d'évaluation de l'utilisabilité.

##### **Conclusions de l'étude :**

Selon les trois critères sus-cités, CW3 et DCW ont fourni des résultats intéressants. En effet, pour le premier critère on a obtenu une moyenne de 1.68 pour DCW contre 0,95 pour CW3. C'est-à-dire que DCW semble inciter les évaluateurs à être plus sévères qu'avec CW3. Pour le deuxième critère, on a obtenu une moyenne de 2.76 pour DCW contre 1.81 pour CW3. Ce qui veut dire que DCW conduit à des résultats plus pertinents que CW3. Enfin, concernant le troisième critère, on a obtenu 2,38 pour DCW contre 1,71 pour CW3. Ce qui signifie que DCW conduit mieux les évaluateurs à trouver les solutions aux problèmes trouvés que CW3.

#### **2.4.2.7 Etude de Jeffries et al. [1991]**

##### **Condition de l'étude :**

Cette expérience avait pour objectif d'une part de faire tester CW2 par un groupe d'évaluateurs et d'autre part de comparer CW2 avec d'autres méthodes d'évaluation (évaluation heuristique, Usability testing et Guidelines). L'interface cible de l'évaluation était l'interface graphique HP-VUE d'un système d'exploitation Unix d'une station de travail HP. C'était une interface qui n'était pas du type « Walk-up-and use ».

Quatre groupes d'évaluateurs composés chacun d'un nombre variable de participants ont été constitués pour participer à l'étude. Le groupe utilisant la méthode évaluation heuristique était composé de quatre spécialistes de l'IHM, celui de la méthode Usability testing était conduit par un

spécialiste des facteurs humains et était composé de six utilisateurs réguliers de PC mais non familiers avec le système UNIX. Enfin ceux exploitant d'une part Guidelines et d'autre part CW2 étaient composés chacun de trois informaticiens qui ne faisaient pas partie des concepteurs du système mais étaient membres du laboratoire d'appartenance.

#### **Conclusions de l'étude :**

A l'issue des évaluations, la méthode évaluation heuristique a permis d'identifier 105 problèmes d'utilisabilité, usability testing 31, guidelines 35 et enfin CW2 35. [Jeffries et al., 1991] ont constaté que même si la méthode d'évaluation heuristique permet d'identifier de nombreux problèmes, en particulier les problèmes sérieux, elle nécessite la participation de spécialistes de l'IHM. Tandis que CW2 peut être utilisé par des non spécialistes de l'IHM en particulier les développeurs.

### **2.4.2.8 Etude de John et Marks [1997]**

#### **Conditions de l'étude :**

L'objectif était d'évaluer la puissance prédictive, de même que la puissance persuasive de six méthodes d'évaluation (CW3, Heuristic Evaluation (HE), Claims Analysis, GOMS, User Action Notation, et par lecture de la spécification (« simply reading the specification »)) en comparant leurs prédictions à celle de tests utilisateur (pensée à haute voix). Six évaluateurs ont évalué un système de création multimédia éducatif. Ils ont utilisé chacun une méthode différente. L'évaluateur qui a utilisé Claims Analysis avait le niveau licence en ingénierie électrique. Celui qui a utilisé CW3 avait le niveau de maîtrise en informatique. Celui qui a utilisé GOMS avait un niveau maîtrise en architecture. Celui qui a utilisé Heuristic Evaluation avait le niveau maîtrise en anglais. Celui qui a utilisé User Action Notation avait le niveau licence en informatique. Enfin, celui qui a utilisé la lecture de la spécification était en troisième année de licence en informatique. En ce qui concerne la méthode « pensée à haute voix », 20 étudiants en licence de gestion (business) y avaient participé.

#### **Conclusions de l'étude :**

CW3 a permis de détecter 42 problèmes contre 17 pour User Action Notation, 24 pour Claims Analysis, 44 pour GOMS, 69 pour Reading Specification et 88 pour HE. Par contre 2 (5%) des 42 problèmes trouvés par CW3 sont des fausses alarmes (c'est-à-dire qui ne sont pas de problèmes d'utilisabilité) contre 4 (6%) pour Reading Specification, 6 (13%) pour GOMS, 24 (17%) pour HE, 6 (24%) pour Claims Analysis 14 (82%) pour UserAction Notation.

Parmi les 40 problèmes trouvés à l'aide de CW3, 13 (33%) concernent des aspects du système qui ne sont pas encore implémentés, contre 5 sur 18 (26%) pour Claims Analysis, 17 sur 38 (44%) pour GOMS, 3 sur 64 (5%) pour HE, 3 sur 3 (100%) pour User Action Notation, et 20 (31%) pour Reading Specification. Ce qui confirme ce qu'ont dit Wharton et al. (1994) : CW peut être utilisée tôt dans le processus de développement.

En plus, parmi les 15 problèmes trouvés par CW3 concernant les aspects du système qui sont déjà implémentés 11 sur 15 (73%) conduisent les développeurs à effectuer des modifications sur le code, contre 0 sur 11 (0%) pour Claims Analysis, 3 sur 10 (30%) pour GOMS, 7 sur 41 (17%) pour HE, 0 sur 0 (0%) pour User Action Notation, et 9 sur 23 (39%) pour Reading Specification.

En combinant les problèmes similaires, le nombre de problèmes trouvés par les six évaluateurs sont réduits à 54. Parmi ces 54 problèmes, 26 étaient observés dans les tests utilisateurs. Notons que parmi ces 26 problèmes, 13 ont suggéré le changement de code

Tableau 2.2. Synthèse des études comparatives

CW1 = Cognitive Walkthrough, first version  
CW2 = Cognitive Walkthrough, second version  
CW3 = Cognitive Walkthrough, third version  
HW = Heuristic Walkthrough  
NCW = Norman's Cognitive Walkthrough  
SCW = Streamlined Cognitive Walkthrough  
CWW = Cognitive Walkthrough for the Web  
GW = Groupware Walkthrough  
AW = Activity Walkthrough  
IW = Interaction Walkthrough  
CWU = Cognitive Walkthrough with Users  
TCW = exTended Cognitive Walkthrough  
DCW = Distributed Cognitive Walkthrough  
HCW = enHanced Cognitive Walkthrough  
ME = Méthode empirique  
TA = Think Aloud  
UT = Usability testing  
GL = Guideline  
CA = Claims Analysis  
GOMS = Goals, Operators, Methods and Selection rules  
UAN = User Action Notation  
RS = Reading specification  
.



Etude	CW			variantes de CW										Autres moyens d'évaluation										Type de système évalué	Type et nombre d'évaluateurs	Résultats numériques concernant la détection de problèmes d'utilisabilité		
	CW1	CW2	CW3	HW	NW	SW	CW	GW	AW	IW	CW	TW	DW	HW	ME	TA	HE	UT	GL	CA	GOMS	UANS	RS					
Lewis et al [1990]	X														X											Système de messagerie	4 experts	Détection avec CW1 de 50% de problèmes trouvés par la méthode empirique
Polson et al [1992]		X														X										Système de messagerie vocal,	Etudiants (nombre non fourni)	Détection avec CW2 de 50% de problèmes trouvés par think aloud
		X													X											Système de traitement de texte,	Etudiants (nombre non fourni)	Détection par CW2 de 30% de problèmes trouvés par d'autres méthodes (interview utilisateur, field trouble report)
		X																	X								« Document routing »	Etudiants (nombre non fourni)
Sears [1997]			X	X																						Système d'apprentissage des effets visuels des algorithmes de restitution d'image.	20 étudiants	Pour l'évaluation heuristique : - en moyenne 1,5 à 3 problèmes sérieux selon le nombre d'évaluateurs. - en moyenne 2,5 à 5,8 problèmes de sévérité intermédiaire selon le nombre d'évaluateurs. - en moyenne 3 à 10,7 problèmes de sévérité mineure selon le nombre d'évaluateur. - en moyenne 1,7 à 7 faux problèmes selon le nombre d'évaluateurs

Etude	CW			variantes de CW									Autres moyens d'évaluation									Type de système évalué	Type et nombre d'évaluateurs	Résultats numériques concernant la détection de problèmes d'utilisabilité					
	CW1	CW2	CW3	HW	NW	SW	CW	GW	AW	IW	CW	TW	DW	HW	ME	TA	HE	UT	GL	CA	GOMS				UANS	RS			
Sears [1997]			X	X													X										Système d'apprentissage des effets visuels des algorithmes de restitution d'image.	20 étudiants	<p>Pour l'évaluation heuristique :</p> <ul style="list-style-type: none"> <li>- en moyenne 1,5 à 3 problèmes sérieux selon le nombre d'évaluateurs.</li> <li>- en moyenne 2,5 à 5,8 problèmes de sévérité intermédiaire selon le nombre d'évaluateurs.</li> <li>- en moyenne 3 à 10,7 problèmes de sévérité mineure selon le nombre d'évaluateurs.</li> <li>- en moyenne 1,7 à 7 faux problèmes selon le nombre d'évaluateurs (faire trait fin en-dessous)</li> </ul> <p>Pour CW :</p> <ul style="list-style-type: none"> <li>- en moyenne 1,3 à 3 problèmes sérieux selon le nombre d'évaluateurs.</li> <li>- en moyenne 2,9 à 5 problèmes de sévérité intermédiaire selon le nombre d'évaluateurs.</li> <li>- en moyenne 2 à 6 problèmes de sévérité mineure selon le nombre d'évaluateur.</li> <li>- en moyenne 0,3 à 1,4 faux problèmes selon le nombre d'évaluateurs</li> </ul>

Etude	CW			variantes de CW										Autres moyens d'évaluation										Type de système évalué	Type et nombre d'évaluateurs	Résultats numériques concernant la détection de problèmes d'utilisabilité			
	CW1	CW2	CW3	HW	NW	SW	CW	GW	AW	IW	CW	TW	DW	HW	ME	TA	HE	UT	GL	CA	GOMS	UANS	RS						
Sears [1997] (Suite)			X	X													X										Système d'apprentissage des effets visuels des algorithmes de restitution d'image.	20 étudiants	Pour heuristic walkthrough: - en moyenne 2,1 à 3 problèmes sérieux selon le nombre d'évaluateurs. - en moyenne 3,3 à 5,9 problèmes de sévérité intermédiaire selon le nombre d'évaluateurs. - en moyenne 3,4 à 9,6 problèmes de sévérité mineure selon le nombre d'évaluateurs. - en moyenne 0,3 à 1,4 faux problèmes selon le nombre d'évaluateurs
Granoller & Lorés [2004]			X														X										Simulateur de microscope	Un expert et 10 étudiants	Détection avec CW3 de 31% des problèmes trouvés par CW with Users
																											Système de gestion de conférence	Un expert et 3 utilisateurs	Détection avec CW3 de 37% des problèmes trouvés par CW with users
Kato & Hori [2005]			X																								Interface d'un appareil photo numérique	20 étudiants	Détection avec CW3 de 72% des problèmes trouvés par Extended CW
																											Logiciel de dessin.		Détection avec Extended CW de 88% des problèmes trouvés par CW3
Eden [2008]			X																								Système de restauration	42 étudiants en informatique	Critère « average severity rating » =1.68 pour DWC contre 0.82 pour CW ; « actionability rating » =2.38 pour DCW contre 1.71 pour CW, « usability relevance rating » =2.76 pour DCW contre 1.81 pour CW

Etude	CW			variantes de CW										Autres moyens d'évaluation								Type de système évalué	Type et nombre d'évaluateurs	Résultats numériques concernant la détection de problèmes d'utilisabilité			
	CW1	CW2	CW3	HW	NW	SW	CW	GW	AW	IW	CW	TW	DW	HW	ME	TA	HE	UT	GL	CA	GOMS				UANS	RS	
Jeffries et al [1991]		X															X	X	X						Interface graphique d'un système d'exploitation	3 informaticiens (utilisaient CW2), 4 spécialistes d'IHM (évaluation heuristique), un professionnel des facteurs humains et 6 utilisateurs (Usability testing), 3 autres informaticiens (Guidelines)	Détection avec CW2 de 35 problèmes, 105 avec l'évaluation heuristique, 31 avec l'Usability testing et 35 en exploitant Guideline
John & Marks [1997]			X														X			X	X	X	X	Système de création multimédia	Informaticien (Master degree)	Détection avec CW3 de 42 problèmes dont 2 faux problèmes, 13 problèmes sur les aspects non implémentés du système et 11 problèmes qui conduisent les développeurs à modifier le code.	
																									Évaluateur ayant une maîtrise en anglais et des compétences dans 2 langages de programmation	Détection avec l'évaluation heuristique de 88 problèmes dont 24 faux problèmes, 3 problèmes sur les aspects non implémentés du système et 7 problèmes qui conduisent les développeurs à modifier le code.	
John & Marks [1997] (Suite)			X														X			X	X	X	X	Système de création multimédia	Évaluateur ayant une maîtrise en architecture et des compétences dans 3 langages de programmation	Détection avec GOMS de 44 problèmes dont 6 faux problèmes, 17 problèmes sur les aspects non implémentés du système et 3 problèmes qui conduisent les développeurs à modifier le code. (faire trait fin en-dessous)	



## 2.4.3 Synthèses sur les études non comparatives

Le tableau 2.3 récapitule l'ensemble des études non comparatives menées par divers auteurs. Pour chaque étude, le système évalué est précisé, de même que les types de sujets ayant participé à l'étude. Quand des résultats numériques sont disponibles, ils sont fournis.

### 2.4.3.1 Etude de Rizzo et al. [1997]

#### Conditions de l'étude :

Les auteurs n'ont pas présenté d'étude avec résultats numériques, ils se sont contentés de décrire un exemple mettant en œuvre la méthode. Le scénario étudié impliquait une personne avec chaise roulante qui vivait à Rome et décidait de planifier une visite d'un jour à Sienne, en Italie. Pour chaque activité de la tâche les membres de l'équipe de conception et d'évaluation ont appliqué les questions du modèle d'action de manière répétitive.

#### Conclusions de l'étude :

La conclusion de l'étude, relativement à la méthode Norman's CW, est la suivante : « La méthode Norman's CW a permis d'atténuer les problèmes de conception que rencontrent des équipes de conception travaillant dans des endroits différents ».

### 2.4.3.2 Etude de Spencer [2000]

#### Conditions de l'étude :

Cette étude avait pour objectif de mettre en évidence l'efficacité de la méthode Streamlined CW dans un contexte de conduite de projet sous contrainte sociale (cf § 2.3.3). Pour cela, un « Integrated Development Environment » (IDE) en cours de développement a été évalué par une équipe de 8 personnes (3 spécialistes de l'utilisabilité, 1 spécialiste du graphisme, et 4 chefs de projet responsables de différents aspects des spécifications de l'IDE). Seul le chef de groupe, un spécialiste de l'utilisabilité était familier de la méthode CW. Le test s'est déroulé en deux sessions espacées d'une semaine. La première session a duré 90 minutes dont 20 minutes consacrées à la phase de préparation. Elle a couvert 32 séquences d'actions.

#### Conclusions de l'étude :

La première session a permis de trouver 24 problèmes potentiels. Les participants se sont mis d'accord sur les faits que : 14 des 24 problèmes d'utilisabilité trouvés étaient dûs aux insuffisances de connaissances de la part de certains utilisateurs, et que les 10 qui restent sont dûs au fait que le système ne fournissait pas de feed-back aux utilisateurs quand l'action correcte a été réalisée. Les résultats montrent que l'équipe n'a pas perdu de temps à discuter pour défendre leurs choix de conception.

### 2.4.3.3 Etude de Blackmon et al. [2002]

#### Conditions de l'étude :

Cette étude visait à comparer le pourcentage de réussite des premiers clics des participants sur des liens ou textes affectés de problèmes d'utilisabilité et celui des liens ou textes sans problèmes. Pour cela, ils ont effectué trois expériences sur la simulation de recherche d'information dans une encyclopédie en ligne. Les participants à ces trois expériences étaient de niveau

universitaire (College educated). L'expérience 1 a été effectuée sur une première encyclopédie en ligne. L'expérience 2 a été effectuée sur la même encyclopédie, mais avec un affichage supplémentaire (une description longue du but en cours). L'expérience 3 a été effectuée sur une seconde encyclopédie en ligne. Elle utilisait la même procédure que l'expérience 2 (c'est-à-dire avec un affichage supplémentaire relatif au but en cours).

#### **Conclusions de l'étude :**

Une partie des résultats de l'analyse est montrée dans le tableau 2.3. On constate que le taux de réussite de l'expérience 2 était supérieur à celui de l'expérience 1 (environ 23 points de différence). Puis le taux de réussite de l'expérience 3 est inférieur à celle de l'expérience 2 (environ 29 points de différence). Selon Blackmon et al. [2002], les résultats montrent que « the CWW can identify characteristics of web pages that differentially affect user performance ». En plus CWW fournit des diagnostics spécifiques qui peuvent guider la réparation des problèmes qu'elle identifie.

### **2.4.3.4 Etude de Pinelle et Gutwin [2002]**

#### **Conditions de l'étude :**

Il s'agit d'une étude de cas que les auteurs ont réalisée pour montrer comment utiliser GW. Pour cela ils ont évalué un premier prototype d'un système de soin à domicile. Le système permet aux membres de l'équipe médicale (docteurs, infirmières, etc.) de communiquer et de partager des documents. La communication est réalisée à travers un outil de bavardage.

#### **Conclusions de l'étude :**

A l'issue de l'étude de cas, cinq principaux problèmes d'utilisabilité ont été détectés : le récepteur d'un message peut ne pas détecter sa présence, la variabilité des compétences de frappe des utilisateurs peut entraîner des difficultés, le système ne fournit pas assez d'information sur la disponibilité d'une personne, le système a des lacunes en termes de rencontres multiples dans des sous-groupes d'utilisateurs, l'identification d'un élément sur un document partagé peut être source de problèmes. Ils ont constaté que GW les a conduits à réviser facilement et rapidement leur conception.

### **2.4.3.5 Etude de Huart et al. [2004]**

#### **Conditions de l'étude :**

Cette expérimentation concerne CW2. Elle se focalise sur l'évaluation de quatre systèmes multimédia de différents degrés d'interactivité. Le premier système est une encyclopédie pour enfant de 8 à 14 ans, c'est le système de plus bas degré d'interactivité de l'étude. Le second système est un système d'exploration d'un musée destiné à tout public. Le troisième système est un système ludo-éducatif destiné aux enfants à partir de quatre ans. Le quatrième système est un système de divertissement à réalité virtuelle. C'est le système de plus haut degré d'interactivité de l'étude. Quatre évaluateurs, dont un spécialiste de l'ergonomie, un spécialiste de la conception et de l'évaluation de système interactif (appelés évaluateurs experts) et deux ergonomes moins expérimentés, étudiants en master d'ergonomie industrielle (appelés évaluateurs non experts), ont évalué successivement les quatre systèmes.

#### **Conclusions de l'étude :**

A l'issue de l'expérimentation, 80 problèmes ont été détectés pour les quatre systèmes dont 22 ont été détectés par au moins deux évaluateurs. Parmi les problèmes détectés par les experts, 58

ne l'ont pas été par les non-experts ; parmi ceux détectés par les non-experts, 6 ne l'ont pas été par les experts. Huart et al. [2004] ont conclu que CW2 peut être utilisée pour évaluer des systèmes multimédia de différents niveaux d'interactivité. Néanmoins, l'efficacité de la méthode dépend fortement de l'expertise de l'évaluateur : dans l'étude, les évaluateurs plus expérimentés ont trouvé beaucoup plus de problèmes que ceux moins expérimentés.

### **2.4.3.6 Etude de Bertelsen [2004]**

#### **Conditions de l'étude :**

L'objectif de cette étude préliminaire était d'obtenir des informations sur l'applicabilité et l'utilité de AW. Les sujets étaient des étudiants (niveau non précisé dans l'étude). Ceux-ci suivaient le cours d'IHM de l'auteur se constituaient en trinôme ; chaque trinôme a évalué une interface homme-machine de leur choix.

#### **Conclusions de l'étude :**

Cette étude préliminaire indique que AW est très compliquée à utiliser dans sa forme actuelle. Néanmoins, plusieurs étudiants ont conclu que la contextualisation dans la phase 3 est une avancée importante si on se réfère à l'état de l'art des méthodes d'inspection.

### **2.4.3.7 Etude de Ryu et Monk [2004]**

#### **Conditions de l'étude :**

Les auteurs se contentent de décrire les étapes et les questionnaires nécessaires à l'utilisation de la méthode IW. Ils n'ont pas proposé d'étude réelle et complète, mais illustrent leurs explications d'extraits d'interfaces homme-machine.

#### **Conclusions de l'étude :**

Les auteurs ont conclu que la méthode IW est une alternative prometteuse d'analyse des interactions de bas niveau dans un IHM.

### **2.4.3.8 Etude de Bligard et Osvalder [2007]**

#### **Conditions de l'étude :**

Dans cet article, Enhanced CW était intégrée dans une approche plus large de détection de problèmes d'utilisabilité et d'erreurs d'utilisation. Deux études ont été menées sur une interface d'une machine de dialyse (dans le domaine médical). Dans la première étude, l'approche proposée par les auteurs était utilisée pour comparer trois différentes solutions de conception. Tandis que dans la seconde étude, l'objectif était de déboucher sur l'amélioration d'une conception.

#### **Conclusions de l'étude :**

Les auteurs considèrent que les évaluations ont permis d'apporter plus d'informations spécifiques que n'en possédait la compagnie concernant les erreurs d'utilisation et les problèmes d'utilisabilité du prototype de la machine de dialyse. Mais les résultats fournis ne montrent aucune information concernant la performance de Enhanced CW si on l'utilise en dehors de l'approche analytique de prédiction et d'identification de problèmes d'utilisabilité et d'erreurs d'utilisation.



Tableau 2.3 : Synthèse des études non comparatives

Etude	CW			variantes de CW											Type de système évalué	Type et nombre d'évaluateurs	Conclusion de l'étude	
	C W 1	C W 2	C W 3	H W	N C W	S C W	C W W	G W	A W	I W	C W U	T C W	D C W	H C W				
Rizzo et al. [1997]					X											Système de planification de déplacement des personnes âgées et handicapées	Membres de l'équipe de conception et d'évaluation	La méthode a permis à l'équipe travaillant dans différents laboratoires d'éviter l'ambiguïté dans la communication des problèmes trouvés pendant l'évaluation.
Spencer [2000]						X										Integrated Development Environment (IDE)	3 spécialistes de l'utilisabilité, 1 spécialiste du graphisme, et 4 chefs de projets	Détection de 24 problèmes potentiels en 90 mn.
Blackmon et al. [2002]							X									Site web (encyclopédie en ligne)	Etudiants (nombre varié selon les expériences)	<p><b>Expérience 1</b> : 41% de taux de réussite de premier click pour les buts affectés de problèmes d'utilisabilité détectés par CWW contre 70% pour les buts sans problèmes.</p> <p><b>Expérience 2</b> : 68% de taux de réussite de premier click pour les buts affectés de problèmes d'utilisabilité détectés par CWW contre 89% pour les buts sans problèmes.</p> <p><b>Expérience 3</b> : 38% de taux de réussite de premier click pour les buts affectés de problèmes d'utilisabilité détectés par CWW contre 62% pour les buts sans problèmes.</p>
Pinelle & Gutwin [2002]								X								Système de soin à domicile	Les 2 auteurs	Cinq principaux types de problèmes détectés

Etude	CW			variantes de CW											Type de système évalué	Type et nombre d'évaluateurs	Conclusion de l'étude
	C W 1	C W 2	C W 3	H W	N C W	S C W	C W W	G W	A W	I W	C W U	T C W	D C W	H C W			
Huart et al. [2004]		X													Application multimédia de type : encyclopédie pour enfant (degré d'interactivité 1)	2 experts : 1 personne spécialiste de l'utilisabilité, et un informaticien, et 2 ergonomes novices	Les évaluateurs experts ont trouvé 18 problèmes contre 9 pour les évaluateurs novices.
Huart et al. [2004] (Suite)		X													Application multimédia de type : système de visite interactive de musée (degré d'interactivité 2)	2 experts : 1 spécialiste de l'utilisabilité, et un informaticien, et 2 ergonomes novice	Les évaluateurs experts ont trouvé 16 problèmes contre 5 pour les évaluateurs novices.
			Application multimédia du type : logiciel ludo-éducatif destiné aux enfants à partir de quatre ans (degré d'interactivité 3)	Les évaluateurs experts ont trouvé 20 problèmes contre 5 pour les évaluateurs novices.													
			Application multimédia du type : jeu en réalité virtuelle pour tout public (degré d'interactivité 4)	Les évaluateurs experts ont trouvé 21 problèmes contre 4 pour les évaluateurs novices.													
Bertelsen [2004]									X					Projets d'étudiant	Etudiants	Méthode considérée comme trop compliquée pour les étudiants qui l'ont testée	
Ryu & Monk [2004]										X				Aucune (seules des illustrations sont fournies dans l'article)	Sans objet	Aucune donnée numérique n'est fournie, mais des pistes intéressantes sont discutées sous l'angle d'une aide que la méthode peut apporter par rapport à la conception.	
Bligard & Osvalder [2007]													X	Machine de dialyse	Non communiqué	Aucune donnée numérique n'est fournie permettant de tirer des conclusions sur la validité de HCW, néanmoins l'approche globale semble efficace relativement à l'évaluation de systèmes complexes (du type évalué).	

## 2.4.4 Synthèse en vue d'une première aide au choix d'une version ou variante

Nous n'avons trouvé aucun document visant à aider les évaluateurs (chercheurs et praticiens) à choisir une méthode parmi les différentes versions et variantes de CW présentées dans la littérature. Cela semble *a priori* une tâche extrêmement difficile (voire impossible) au vu des différences très importantes entre les documents étudiés : structure, manière dont les expériences ont été menées, types de résultats attendus dans les études, niveau d'avancement des versions et variantes. Les différentes synthèses (§ 2.4.1, § 2.4.2, § 2.4.3) que nous avons proposées pourraient constituer une toute première aide à ce sujet.

Nous essayons de faire un pas de plus dans cette direction: le tableau 2.4 a pour objectif de compléter les différentes synthèses précédentes. Nous y mettons en évidence quelques citations clés dans l'article original, que nous jugeons caractériser la spécificité de la méthode. Enfin, nous proposons pour chaque version et variante un avis informel.

Tableau 2.4 : Citations et perspectives sur la spécificité des diverses versions et variantes

CW, Variants and evolutions	Citation	Point de vue
<b>CW1</b>	« the walkthrough with a very limited investment in resources, approximately an hour per task per interface, can detect almost 50 percent of the problems encountered by users of the design » [Lewis et al., 1990, page 240].	Prévu pour les interfaces du type <sup>a</sup> walk-up-and-use <sup>o</sup> même si on n'a que des maquettes. Évalue le pourcentage des utilisateurs potentiels qui peuvent rencontrer de problème pour chaque action
<b>CW2</b>	« The cognitive walkthrough is an evaluation methodology that focuses on ease of learning. It is especially appropriate for the development of applications where users must (or prefer) to master a new application or function by learning through exploration » [Polson et al., 1992, page 742].	Prévu pour les interfaces du type <sup>a</sup> walk-up-and-use <sup>o</sup> même si on n'a que des maquettes. Évalue le pourcentage des utilisateurs potentiels qui peuvent rencontrer de problème pour chaque action
<b>CW3</b>	« The cognitive walkthrough method promises to be a valuable addition to the designer's suite of tools. The new version of a method is flexible enough to fit into given software development process. The method identifies problems with a design early in the process and, by describing the reasons for those problems, it suggests design changes early on. » [Wharton et al. 1994, page 139].	Justifie le choix de l' action par de scénario
<b>HW</b>	« A new techniques is described that combines the benefits of heuristics evaluation, cognitive walkthroughs, and usability walkthrough » [Sears, 1997, page 243].	C' est une méthode qui associe des approches bien connues par les professionnels de l' utilisabilité, en particulier l' évaluation heuristique. Ceci lui confère la possibilité de détecter potentiellement plus de problèmes d' utilisabilité.

<b>CW, Variants and evolutions</b>	<b>Citation</b>	<b>Point de vue</b>
<b>NCW</b>	« The design issues are: the problems deriving from distribution of the teams collaborating to the project in several cities (sometimes different European countries); and the need to face high-level interaction problems in the evaluation process. One important action taken to face these issues was the development of a variation of the Cognitive Walkthrough based on the Norman©model of action. » [Rizzo et al., 1997, page 305].	C'est une méthode qui permet de résoudre plutôt des problèmes de haut niveau (au contraire de IW, cf. ci-dessous). C'est intéressant pour une première approche d'évaluation d'un système interactif, sans entrer dans les détails de sa réalisation.
<b>SCW</b>	« Managers, developers, and other team members are pressured for time, tend to lapse into lengthy design discussions, and are sometimes defensive about their user-interface designs. By enforcing four ground rules, explicitly defusing defensiveness, and streamlining the cognitive walkthrough method and data collection procedures, these social constraints can be overcome, and useful, valid data can be obtained. This paper describes a modified cognitive walkthrough process that accomplishes these goals » [Spencer, 2000, page 353]	Cette variante apporte une dimension sociale qui n'existe pas dans les autres variantes.
<b>CWW</b>	« The new cognitive walkthrough for the Web (CWW) is superior for evaluating how well websites support users' navigation and information search task » [Blackmon et al., 2002, page 463]	C'est une méthode qui permet de critiquer successivement la conception de pages web en cours de conception ou d'évaluer les pages d'un site en ligne. Cette méthode est en plus automatisée, ce qui n'est pas le cas pour toutes les autres.
<b>GW</b>	« The technique is a substantive modification of cognitive walkthrough to include consideration for the complexities of teamwork. The two components of groupware walkthrough are a task model for identifying and analysing real-world collaborative tasks, and a walkthrough process for assessing a system's support for those tasks. » [Pinelle and Gutwin, 2002, page 455]	C'est une méthode d'inspection des tâches qui utilise GTA. Elle est appropriée pour l'évaluation à l'aide de scénarios collaboratifs.
<b>AW</b>	« The method is a modified version of the cognitive walkthrough, and is aimed to systematically include the context and history of use. » [Bertelsen, 2004, page 251]	C'est une méthode basée sur la théorie de l'activité qui intègre systématiquement le contexte d'utilisation ce qui n'est pas le cas pour les autres variantes. L'avantage de cette méthode est qu'elle permet un compromis : les contextes sont pris en compte, sans une nécessité de les détailler en profondeur.
<b>IW</b>	« The method is a modified version of cognitive walkthrough and the analysis focuses on the issue of direct concern to the practitioner who intends to identify low-level interaction problems in their design specification. » [Ryu et Monk, 2004, page 304]	Cette méthode présente une alternative d'analyse des interactions homme-ordinateur de bas niveau.

<b>CW, Variants and evolutions</b>	<b>Citation</b>	<b>Point de vue</b>
<b>CWU</b>	« The new variant which can be regarded as a new method incorporates users in way that combines the advantages of the initial method with those contributed by the availability of end users in the evaluation of interactive systems. » [Granollers et Lores, 2006, page 254]	C'est une méthode qui combine à la fois l'intervention d'experts et d'utilisateurs, au contraire de toutes les autres versions et variantes. C'est une démarche intéressante qui intègre la verbalisation pensée tout haut (think aloud).
<b>Extended CW</b>	« Applied to a Web design evaluation study, the extended CW was shown to be more effective in identifying accessibility and usability problems while remaining as efficient as the currently-practiced CW. » [Kato et Hori, 2007, page 1]	C'est une méthode intéressante pour les évaluateurs possédant peu de connaissance en psychologie cognitive. Elle est prévue en outre pour détecter plus facilement les problèmes d'accessibilité.
<b>DCW</b>	« The Distributed Cognitive Walkthrough (DCW) method is useful for the identification of aspects of usability issues related to interaction between people, artifacts, and information, across dimensions such as time, space, and social structures. » [Eden, 2008, page 2] « DCW method uses concepts from distributed cognitive theory to view interaction between people and information as transcending interactions with graphical user interfaces, allowing the DCW method to be useful for evaluation of design ideas in almost all areas of interaction design; for example, evaluation of ubiquitous computing, service design (e.g. Starbucks customer/worker experience), and mathematical notations (e.g. Newton versus Leibniz Calculus notation). » [Eden, 2007, page 1]	DCW offre des perspectives intéressantes dans le cadre des interactions en général, c'est-à-dire pas seulement entre un humain et un système interactif. Elle est basée sur la cognition distribuée, ce qui n'est pas le cas dans les autres
<b>Enhanced CW</b>	« The approach is based on the methods Hierarchical Task Analysis (HTA), Enhanced Cognitive Walkthrough (ECW) and Predictive Use Error Analysis (PUEA). 3» [Bligard, 2007, page 427] « ECW employs a clearly detailed procedure for simulating the user's problem-solving process in each step of the interaction. Throughout the interaction, it is checked whether the supposed user's established goal and previous experience will lead to the next correct action » [Bligard, 2007, page 432]	C'est une méthode qui classe les problèmes selon leur sévérité en proposant 5 classes de problèmes pour une meilleure identification. En plus elle s'intègre dans une méthode plus globale orientée vers les systèmes complexes.

## 2.5 Conclusion

D'une manière générale, CW permet d'évaluer la facilité avec laquelle un utilisateur réalise une tâche avec un minimum de connaissance du système ainsi que la facilité d'apprentissage d'interface par l'exploration. A l'origine, elle est basée sur le modèle d'apprentissage de [Polson et al. 1990], lui-même inspiré de la théorie de l'action de Norman [Norman, 1994]. Les utilisateurs intègrent une entrée perceptive avec leurs connaissances de base pour se construire une

représentation qui va leur permettre de réaliser une tâche. Ainsi CW a pour but de simuler le cheminement cognitif de l'utilisateur le long d'une interface et avec un effort pour accomplir la tâche. Des formulaires avec des questions spécifiques guident le cheminement de l'utilisateur. CW est applicable dans les phases de développement et conception du système, pour déceler les problèmes liés à l'utilisabilité du système dès sa modélisation et la spécification de ses fonctionnalités. Elle peut être utilisée également *a posteriori* pour déterminer les difficultés d'utilisation d'un système lors de la réalisation des scénarios spécifiques et de mise en situations particulières. La représentation des processus cognitifs développés par l'utilisateur permet d'améliorer l'utilisabilité.

CW est une méthode d'évaluation largement reconnue dans la communauté scientifique des IHM. Ses fondements théoriques associés à son intérêt pratique font qu'elle a, depuis son apparition, fait l'objet de nombreuses études et extensions, sous les angles aussi bien conceptuels, méthodologiques, que technologiques.

D'après l'analyse détaillée de la bibliographie, on ne trouve pas encore d'environnement qui couvre les différentes phases d'une session d'évaluation par la méthode CW. Ce qui nous amène à proposer un environnement d'aide à l'évaluation des systèmes interactifs basée sur la méthode Cognitive Walkthrough dont la spécification est décrite dans le chapitre 3 qui suit.

# Chapitre 3 - Spécification d'un environnement d'évaluation de systèmes interactifs basée sur la méthode Cognitive Walkthrough

---

## Plan du chapitre

3.1- Introduction.....	2
3.2- Le processus d'évaluation.....	2
3.2.1 - Les acteurs de l'évaluation.....	3
3.2.2 - Les étapes de l'évaluation.....	3
3.3- Modélisation statique de l'évaluation.....	6
3.3.1 - Le modèle de l'artefact.....	6
3.3.2 - Le modèle de l'utilisateur.....	8
3.3.3 - Le modèle de tâche.....	9
3.3.4 - Le modèle des méthodes.....	11
3.3.5 - Le modèle de résultat.....	21
3.4- Modélisation dynamique de l'évaluation.....	22
3.4.1 - Spécification de l'artefact.....	22
3.4.2 - Modélisation de l'utilisateur.....	25
3.4.3 - Choix de scénarii à évaluer.....	27
3.4.4 - Analyse de tâches.....	28
3.4.5 - Choix de méthode d'évaluation.....	29
3.4.6 - Prise en main de l'évaluation.....	30
3.4.7 - Identification de l'évaluateur.....	31
3.4.8 - Inspection.....	32
3.4.9 - Analyse des résultats.....	33
3.5- Conclusion.....	34

---

### 3.1 Introduction

Ce chapitre est consacré à la spécification d'une méthodologie d'évaluation de systèmes interactifs basée sur la méthode Cognitive Walkthrough. L'objectif est d'arriver à un environnement d'aide à l'évaluation appelé « Cognitive Walkthrough Environment » (CWE). Ce chapitre est composé de trois parties. La première partie traite du processus d'évaluation ; elle correspond en quelque sorte à l'analyse des besoins. La seconde partie décrit le modèle de domaine qui est constitué des différentes entités de l'évaluation ainsi que des relations entre eux. Enfin la troisième partie concerne le comportement dynamique de l'environnement d'évaluation visé. Pour illustrer les différents propos décrits dans ce chapitre, nous allons utiliser la notation UML (Unified Modeling Language).

### 3.2 Le processus d'évaluation

Nous allons décrire dans cette partie l'analyse du processus d'évaluation. Selon Larman [2005], l'analyse met l'accent sur une investigation du problème et des besoins. Nous allons donc décrire ici « qui sont les acteurs de l'évaluation ? », « quelles sont les phases et fonctionnalités composant ce processus ? ». La figure 3.1 résume la réponse à ces questions.

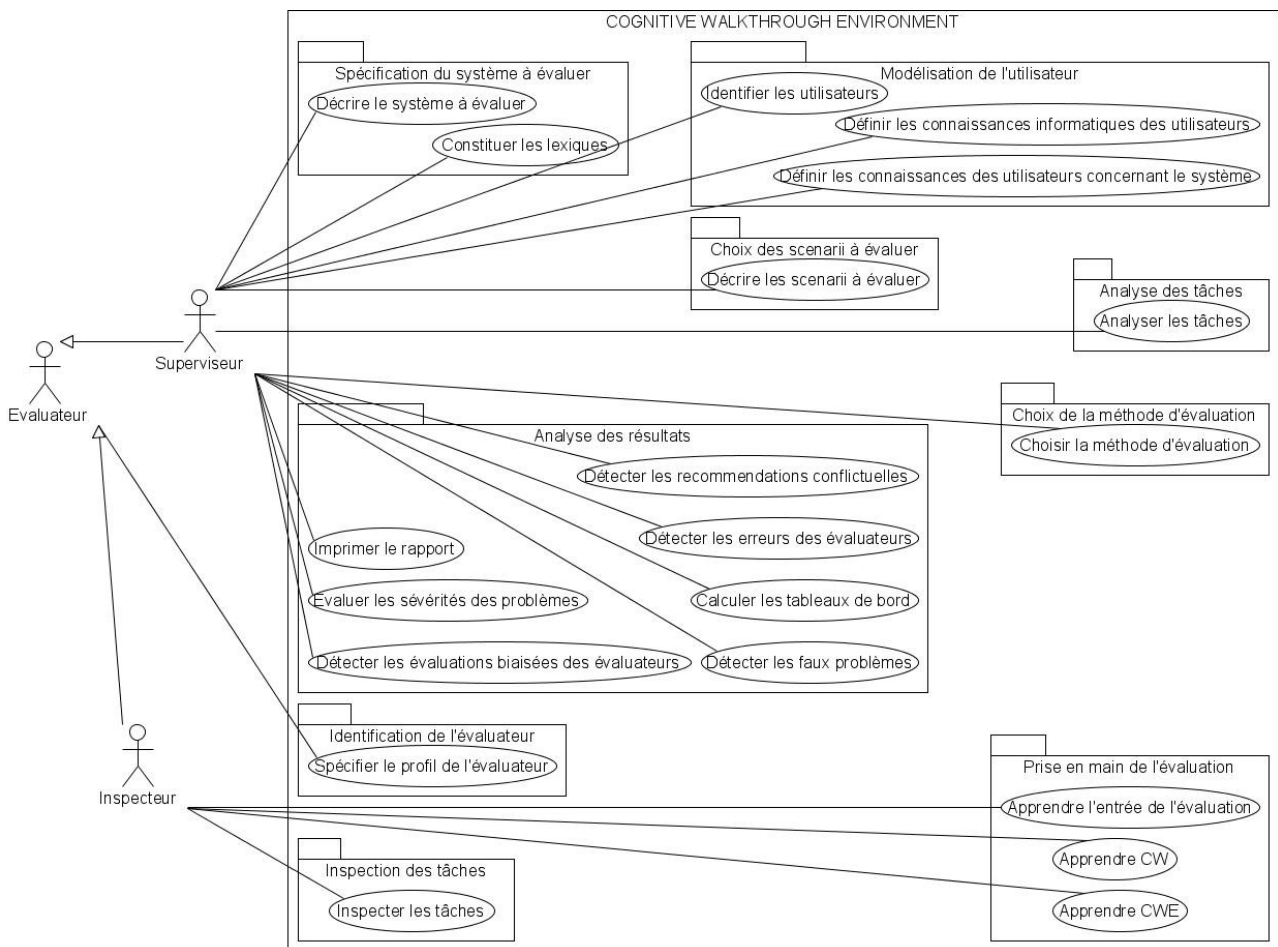


Figure 3.1 : Diagramme de cas d'utilisation



### 3.2.1 Les acteurs de l'évaluation

Chaque individu qui participe à l'évaluation est appelé évaluateur. L'évaluation est donc effectuée par une équipe d'évaluateurs composée d'un ou plusieurs évaluateurs. Les membres de l'équipe peuvent posséder différents profils. En effet, ils peuvent être les développeurs ou les concepteurs qui ont réalisé le système interactif. Mais ils peuvent être aussi des utilisateurs finaux du système. Et enfin ils peuvent être aussi des ergonomes ou spécialistes de l'utilisabilité. Pour la méthode « Streamlined CW » par exemple, l'équipe d'évaluation doit contenir un spécialiste de l'utilisabilité [Spencer, 2000].

A part son profil, chaque évaluateur joue un rôle. Nous rappelons qu'un rôle est un ensemble significatif de tâches exécutées par un ou plusieurs acteurs. Il est caractérisé par un ensemble de tâches et de droits vis-à-vis des autres membres et des données partagées. Nous distinguons deux rôles : le superviseur et l'inspecteur.

Le superviseur est celui qui est à l'origine du processus d'évaluation. Il peut être le concepteur, un spécialiste de l'IHM ou un spécialiste de l'utilisabilité. Dans tout le cas, il doit donc avoir une vision globale et de haut niveau du produit à évaluer. C'est lui qui décide de la tâche à évaluer. Et c'est aussi lui qui tire la conclusion finale de l'évaluation, en se basant sur les évaluations faites par le ou les inspecteurs. Pour la méthode « Streamlined CW », ce sont les membres de l'équipe d'évaluation qui choisissent le superviseur parmi les membres spécialistes de l'utilisabilité [Spencer, 2000].

L'inspecteur a pour rôle de procéder à l'évaluation ou inspection des tâches définies par le superviseur. Il doit utiliser la méthode d'évaluation désignée par le superviseur lors de l'inspection de chaque action. Ce rôle peut être tenu par n'importe quel individu appartenant aux profils cités plus haut. Par contre, ses activités doivent être conformes à celles qu'a demandées le superviseur. Un évaluateur ayant ce rôle doit se mettre à la place des utilisateurs réels.

### 3.2.2 Les étapes de l'évaluation

Pour être conforme à la démarche d'évaluation ergonomique, nous avons réparti les différentes activités du processus d'évaluation en trois phases : la phase de préparation, la phase d'inspection et la phase d'analyse des résultats.

#### 3.2.2.1 La phase de préparation

Avant le processus d'évaluation, le superviseur dispose d'un système prévu pour être utilisé par un ensemble d'utilisateurs. Ce système est par exemple celui qu'un concepteur ou groupe de concepteurs vient de spécifier ou développer. La phase de préparation consiste à définir les entrées de l'évaluation et les contextes d'utilisation du système. Durant cette phase, le superviseur effectuer les activités suivantes : spécification du système à évaluer, modélisation des utilisateurs finaux du système, choix des scénarii à évaluer, analyse des tâches à évaluer et choix de la méthode d'évaluation à utiliser. Ces activités sont successivement expliquées.

##### 3.2.2.1.1 Spécification du système à évaluer

La spécification du système consiste à rassembler les informations concernant l'application à évaluer. Cette activité est d'autant plus nécessaire si les inspecteurs ne font pas partie de l'équipe qui a conçu ou réalisé le système à évaluer. En effet, elle doit fournir aux inspecteurs les

informations leurs permettant d'appropriier les concepts du domaine du système ainsi que le système et son contexte. Considérons l'exemple suivant : dans la phrase « quelle type d'application doit-on utiliser ? » Pour les informaticiens, la réponse peut être StarOffice ou OpenOffice ; car pour eux les applications sont des artefacts qui permettent de réaliser des tâches sur ordinateur. En revanche pour les médecins, l'application est le fait de placer sur le corps certains produits médicamenteux : application vésicante, application de cataplasmes, etc. La spécification du système permet donc de lever les ambiguïtés de ces genres et aussi de donner l'occasion aux inspecteurs de bien connaître les jargons du domaine concerné par le système.

La spécification du système devrait aussi permettre aux superviseurs de choisir la méthode d'évaluation que les inspecteurs devront utiliser ainsi que la façon de conduire l'inspection. En effet, il est clair qu'un système interactif pour un ordinateur de bureau n'aura pas les mêmes caractéristiques ou les mêmes contraintes qu'un système interactif pour des dispositifs portables. Aussi, un système interactif dans une centrale nucléaire ne sera pas évalué de la même manière qu'un site web de commerce électronique.

### **3.2.2.1.2 Modélisation des utilisateurs finaux**

La modélisation des utilisateurs finaux est aussi très importante dans la mesure où la méthode CW préconise à l'origine que les inspecteurs doivent se mettre à la place des utilisateurs réels du système. Elle consiste à rassembler les informations concernant les utilisateurs réels de système. Ces informations devraient renseigner différents aspects comme ceux préconisés par Robert [2003] et Brusilovsky [2007] et en particulier leurs connaissances et leurs expériences. Le superviseur doit donc spécifier les connaissances des utilisateurs sur le domaine concerné par le système. Il faut spécifier aussi s'ils ont déjà utilisé un système similaire à ce qui va être évalué ou éventuellement les précédentes versions du système s'il en existe. Et bien sûr, puisqu'il s'agit de système informatique, Robert [2003] suggère de spécifier aussi les connaissances et expériences des utilisateurs réels en informatique. Avant de commencer l'inspection, les inspecteurs doivent donc consulter ce modèle, ce qui leur facilitera la simulation du cheminement cognitif des utilisateurs et d'imaginer au mieux ce qui pourrait se passer dans les cas réels, c'est-à-dire à chaque étape de l'utilisation du système interactif à évaluer.

### **3.2.2.1.3 Choix des scénarii à évaluer**

Une des contraintes que nous devons respecter est de rester le plus proche possible de la réalité même si les évaluateurs ne sont pas des utilisateurs réels. C'est pourquoi nous adoptons le scénario comme entrée de l'évaluation ; ce qui permet d'évaluer une situation qui soit relativement proche de la réalité. En effet, selon Leite et al. [2002], un scénario décrit les situations qui se produisent dans l'univers de discours. L'univers du discours étant l'ensemble des contextes dans lesquels le système va être utilisé. Il comprend toutes sources d'information et toutes les populations impliquées dans le système. Un scénario est donc une description partielle du comportement du système à un moment et dans un contexte spécifique bien précis.

Le choix des scénarii doit tenir compte du budget et du temps en possession de l'équipe d'évaluateurs ainsi que des importances des différentes tâches supportées par le système. En effet, évaluer l'ensemble des tâches supportées par le système nécessite autant de budget et de temps. Ce qui oblige donc le superviseur à spécifier de scénarii le plus représentatifs possibles, c'est-à-dire qui couvrent le plus possible l'ensemble des fonctionnalités du système. Dans certain cas selon le budget et le temps à leur disposition, il peut décider de n'évaluer que quelques scénarii les plus importants et les plus critiques.

#### **3.2.2.1.4 Analyse des tâches**

L'analyse des tâches permet de décrire et de décomposer les scénarii. La description d'un scénario consiste à définir les informations qui les caractérisent telles que son nom, son but, etc., les contextes et les conditions de son exécution, ainsi que les objets et ressources utilisés par le scénario. La décomposition d'un scénario consiste à le décomposer hiérarchiquement de façon à obtenir une suite d'actions. Les actions composant cette suite seront évaluées les unes après les autres en répondant aux questions de la méthode choisie par les inspecteurs.

Cette activité est particulièrement importante. Les différentes versions et extensions de CW que nous avons vues mentionnent cette activité sans avoir précisé comment on la faire. Nous proposons un modèle comme cadre de conception et une méthode pour la faire.

#### **3.2.2.1.5 Choix d'une méthode d'évaluation**

Le choix de la méthode d'évaluation consiste à désigner la méthode la mieux adaptée que l'inspecteur doit par la suite utiliser. C'est dans cette activité aussi que le superviseur désigne les inspecteurs à effectuer l'évaluation. Le choix de la méthode dépendra du système à évaluer tandis que le choix des évaluateurs dépendra de leurs expériences.

### **3.2.2.2 La phase d'évaluation**

Après la phase de préparation, le système à évaluer est décrit, les profils des utilisateurs réels du système sont connus, les scénarii à évaluer sont connus aussi et décomposés en suite d'actions, et enfin la méthode d'évaluation à utiliser est choisie. Les inspecteurs peuvent donc procéder à l'évaluation. Pour cela, ils doivent effectuer différentes activités de prise en main de l'évaluation et puis ils procèdent à l'exécution de l'évaluation.

#### **3.2.2.2.1 Prise en main de l'évaluation**

Cette activité permet aux inspecteurs de se mettre dans le bain de l'évaluation. CWE doit leur permettre dans un premier temps de s'appropriier les différents termes et concepts du système à évaluer ; ensuite, il s'agit de bien connaître les utilisateurs réels du système et éventuellement d'apprendre la version ou variante de CW qu'ils doivent utiliser.

#### **3.2.2.2.2 Exécution de l'évaluation**

L'exécution de l'évaluation consiste à simuler l'exécution de la tâche à évaluer. A chaque action prévue lors de l'analyse de la tâche, l'évaluateur doit répondre aux questions correspondant à la méthode choisie. L'évaluateur doit justifier la réponse qu'il donne. Et à chaque réponse négative, il doit remplir des fiches problèmes. Il s'agit d'un formulaire qui décrit le problème. Nous allons voir dans la partie trois de ce chapitre comment on décrit le problème.

#### **3.2.2.3 La phase d'analyse des résultats**

La phase d'analyse des résultats consiste à analyser les résultats fournis par les inspecteurs, en vue de produire le rapport d'évaluation. Le superviseur doit d'abord lever les ambiguïtés qui peuvent exister sur les résultats obtenus. Ces ambiguïtés sont d'origines multiples : erreurs des évaluateurs, faux problèmes, évaluations biaisées des évaluateurs, recommandations conflictuelles.

Pour lever ces ambiguïtés, le superviseur doit synthétiser les résultats par différents tableau de bord. Il doit confronter les résultats entre eux, ce qui permet d'expliquer ou prédire certaines hypothèses.

## 3.3 Modélisation statique de l'évaluation

Dans cette partie, nous allons décrire les entités constituant l'évaluation et les relations entre eux. Nous les décrirons à travers des modèles tels que le modèle de l'artefact, le modèle de l'utilisateur, le modèle de tâches, le modèle de méthode, le modèle de résultat. L'ensemble de ces modèles constitue le « modèle de domaine » de notre application (figure 3.2) conformément au processus de développement appelé « Processus Unifié » [Jacobson et al., 1999 ; Kruchten, 2000]. Un modèle de domaine constitue une perspective conceptuelle, non logicielle, des objets du monde réel [Larman, 2005].

### 3.3.1 Le modèle de l'artefact

Nous allons nous baser sur le modèle décrit en §1,2.2 du chapitre 1 pour spécifier ce modèle.

Tout d'abord, il faut souligner qu'un système doit posséder un nom pour le désigner et une description pour pouvoir le connaître sommairement. Pour le site de commerce électronique des 3Belges par exemple, nous pouvons l'appeler 3Belges. Et aussi, nous pouvons le décrire comme suit : un site web de vente par internet de produits de la mode, de produits électroménagers et d'accessoires de décoration.

En outre, un système concerne un domaine que le superviseur doit aussi spécifier. Le domaine correspond au métier de l'application, par exemple le contrôle du trafic aérien. Un domaine possède des lexiques expliquant le ou les jargons utilisés dans le métier. Chaque métier a ses propres termes ; bien entendu les inspecteurs ne sont pas sensés déjà connaître le jargon du système qu'ils doivent évaluer. Certains termes peuvent provoquer des confusions dues à leurs significations différentes d'un métier à un autre. C'est le cas du terme application que nous avons décrit plus haut. D'autres peuvent être totalement inconnus par les inspecteurs. Par exemple, en lisant l'article [Merlin et al., 2007] qui concerne le domaine du contrôle aérien, on a pu découvrir les termes « clairance » et « strip » qui ne sont pas des termes usuels que les inspecteurs sont censés connaître. C'est pourquoi il est important que le superviseur décrive le domaine du système à évaluer et constitue les lexiques du domaine qui vont permettre aux inspecteurs de s'approprier les concepts du système.

Nous avons souligné plus haut qu'un système déployé sur un ordinateur de bureau ne sera pas évalué de la même façon qu'un système déployé sur un téléphone portable. Les contraintes ne seraient pas les mêmes dans les deux cas ; les tailles des écrans ne sont pas les mêmes par exemple. Nous marquons ces différences par l'attribut : contexte système. Nous avons retenu 3 classes de contexte : le contexte embarqué, le contexte portable et les autres contextes ne concernant pas les deux précédents. Le contexte embarqué concerne les interfaces homme-machine intégrées ou incorporées dans le dispositif. C'est le cas des appareils domestiques comme les fours, les machines à laver. Quant au contexte portable, il concerne les interfaces accessibles via des dispositifs portables (téléphones portables, PDA, etc.).

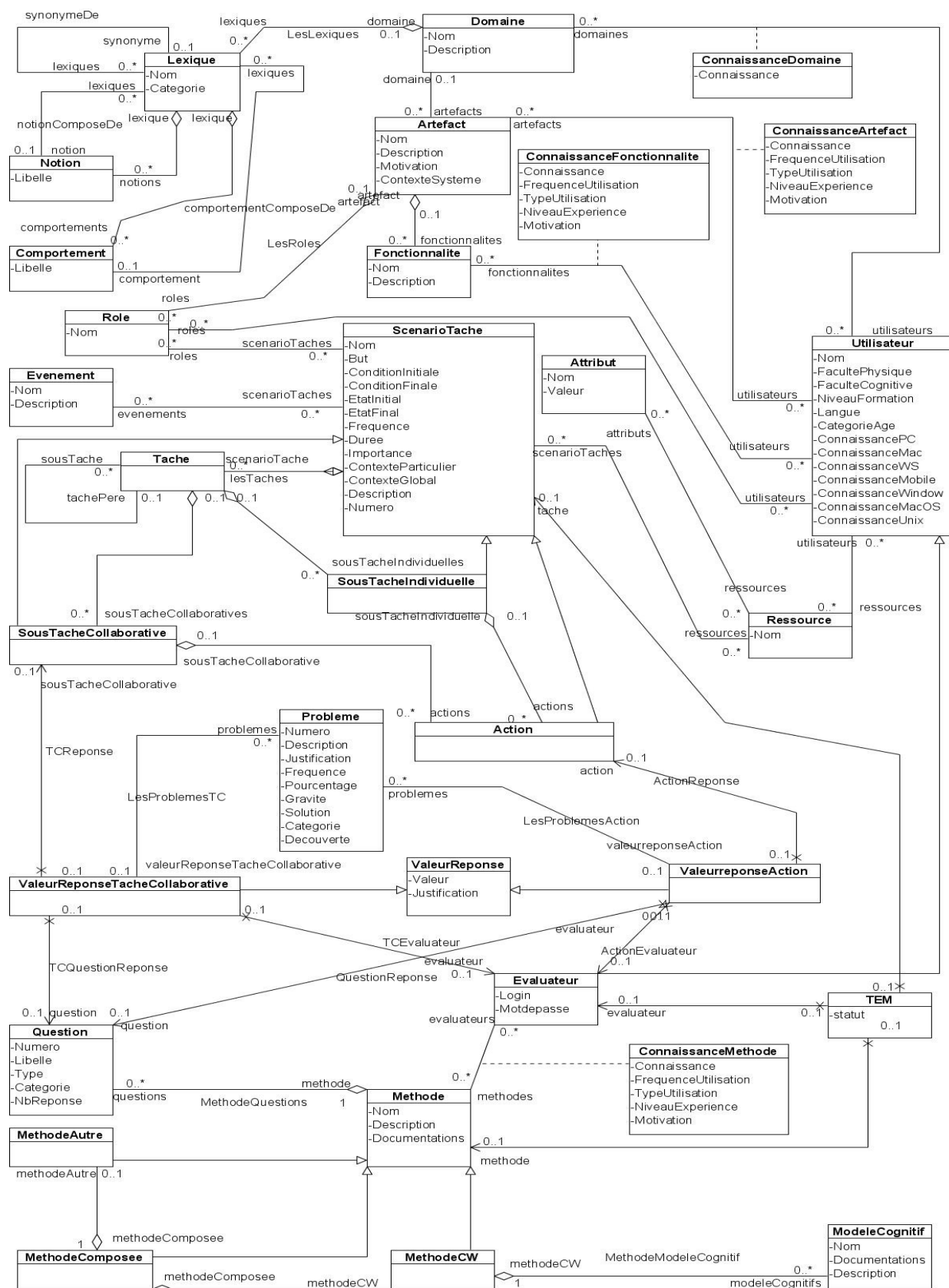


Figure 3.2 : Modèle de domaine

En plus de ces contextes, la motivation de l'application est aussi importante à connaître. Elle peut influencer également le choix de la méthode d'évaluation. Nous avons retenu pour l'instant deux classes de motivation : la motivation industrielle et commerciale et la motivation domestique. Il est important de prendre en compte ces deux motivations parce que les enjeux ne sont pas les mêmes pour les deux cas. Pour la première, la productivité et le coût sont primordiaux, c'est-à-dire que le système a été conçu certainement pour améliorer l'efficacité ou la productivité de l'entreprise tout en maîtrisant les coûts de mise en service et d'exploitation du système. Dans ce cas l'évaluation doit par exemple se focaliser sur la facilité d'utilisation pour réduire le coût de formation sur l'utilisation du système et réduire le taux d'erreurs. Tandis que pour la deuxième motivation, l'esthétique et la facilité d'usage sont très importantes.

Et enfin une application possède des opérations système qui constituent l'interface publique du système [Larman, 2005]. Nous les appelons « fonctionnalité ».

### 3.3.2 Le modèle de l'utilisateur

En ce qui concerne le modèle de l'utilisateur, nous nous basons sur le modèle décrit dans la section §1.4.13 du chapitre 1. Ainsi le modèle comprend pour l'instant les informations suivantes :

- Les connaissances informatiques de l'utilisateur qui peuvent être Faibles, Moyennes ou Bonnes. Ces connaissances concerneront des généralités liées à l'informatique, les plates formes (PC, Macintosh, station de travail) et le système d'exploitation (Unix, Windows, MacOs, Linux),
- Les connaissances de l'utilisateur sur le domaine et le système à évaluer : elles peuvent être aussi Faibles, Moyennes ou Bonnes ; elles concerneront le domaine du système, le système lui-même et d'éventuels systèmes similaires. Considérons par exemple comme système StarOffice, le domaine sera la bureautique et que les systèmes similaires pourront être Microsoft Office ou Openoffice.
- L'expérience de l'utilisateur pourrait être globalement estimée selon trois paramètres à savoir le niveau d'expertise, le type d'utilisation et la fréquence d'utilisation. Le niveau d'expertise peut être de trois types : Novice, Intermédiaire ou Expert. Ensuite le type d'utilisation peut être obligatoire ou discrétionnaire. Et enfin la fréquence d'utilisation peut être faible, moyenne ou élevée. De telles expériences pourront concerner le système et les systèmes similaires.
- Les informations socio-démographiques de l'utilisateur peuvent être de plusieurs ordres : âge (enfant/adolescent, adulte, âgé), sexe (Homme/Femme), langue.
- Les facultés de l'utilisateur (Normales, déficientes) : d'éventuelles déficiences peuvent concerner les membres supérieurs, membres inférieurs, visuelles, auditives.
- La motivation (faible, moyenne, élevée) permet dans la mesure du possible de caractériser l'intérêt de l'utilisateur à utiliser l'application et en particulier à exécuter la tâche.
- Les qualités personnelles des utilisateurs peuvent aider les évaluateurs à prévoir leurs comportements possibles.
- Le rôle de l'utilisateur peut être utile pour préciser la catégorie de celui-ci selon une organisation particulière. Par exemple, Tuteur ou Apprenant sont des rôles dans un système de campus virtuel.

- Le niveau de formation (Primaire, Collège, Lycée, Technicien, Licence, Maitrise, Doctorat) peut s'avérer utile également.

Pour modéliser les connaissances et les expériences de l'utilisateur, nous allons adopter l'approche par superposition [Carr, 1977 ; Polh, 1997 ; Brusilovsky, 2007] qui consiste à utiliser comme cadre de représentation de connaissances le modèle de domaine. Le principe est le suivant : à chaque concept du modèle de domaine, le modèle de connaissance d'un utilisateur mémorise une donnée qui est une estimation de la connaissance de l'utilisateur du concept. Les différents concepts du domaine du système à évaluer sont listés dans les lexiques du domaine. Les connaissances sont évaluées sur le domaine et les fonctions du système.

En ce qui concerne les expériences et les motivations, nous proposons de les évaluer sur l'application et ses fonctions. Ce qui permet de se contenter de l'expérience générale ou de la motivation générale si on n'arrive pas à les déterminer fonction par fonction.

### 3.3.3 Le modèle de tâche

Nous allons utiliser comme base de notre modèle de tâche, le modèle « Group Task Analysis » de Pinelle et Gutwin [2001]. Le choix de ce modèle est motivé par le fait que c'est une méthode d'analyse de tâche en vue de l'évaluation d'une tâche collaborative.

Les principales composantes du modèle de tâche sont les scénarios, les tâches, les sous-tâches individuelles, les sous-tâches collaboratives et les actions. Elles sont indiquées dans la figure 3.2 et décrits ci-dessous.

#### 3.3.3.1 Scénario

Un scénario est une formalisation descriptive du travail que les utilisateurs devront probablement effectuer à l'aide du logiciel. C'est une description de haut niveau des activités liées à l'atteinte d'un résultat précis. Un scénario contient généralement plusieurs tâches et fournit des informations contextuelles sur les utilisateurs et les circonstances dans lesquelles les tâches sont effectuées. Un scénario se compose donc des éléments suivants : une description de haut niveau de l'activité, une spécification des utilisateurs (description des utilisateurs et de leurs connaissances), un résultat (l'objectif visé), et un ensemble de circonstances dans lesquelles le scénario est effectué.

#### 3.3.3.2 Tâche

Les tâches sont des éléments de base du scénario. Elles sont en général explicitement indiquées dans le descriptif du scénario. En revanche, elles décrivent ce qui se passe dans un événement de scénario sans préciser comment cela se produit.

Chaque tâche peut être analysée selon une granularité plus fine et peut être divisée en sous-tâches. Dans le travail de groupe, les tâches peuvent souvent être réalisées avec différents niveaux de couplage entre les membres du groupe. Pour cette raison, les tâches sont analysées en termes de sous-tâches individuelles (faiblement couplés) et sous-tâches de collaboration (étroitement couplées).

##### **Sous-tâches individuelles:**

Les sous-tâches individuelles sont des sous-tâches faiblement couplées. Elles correspondent à des travaux individuels dans une tâche.

### Sous-tâches collaboratives :

Les sous-tâches collaboratives sont des sous-tâches étroitement couplées. Elles correspondent à des travaux collaboratifs dans une tâche. Elles sont effectuées selon une des principes de collaboration proposés par [Gutwin & Greenberg, 2000 ; Pinelle, Gutwin & Greenberg, 2003] (cf. tableau 3.1).

Les mécanismes de collaboration offrent un cadre utile pour spécifier les sous-tâches effectuées en collaboration. Elles fournissent une approche relativement complète pour la capture des différents aspects dans les spécifications de sous-tâche collaborative telles que la communication gestuelle ou la communication verbale.

Le tableau 3.1 donne la liste des principes de collaboration.

**Tableau 3.1 Principes de collaboration**

Catégorie	Principe	Type d'action
Communication explicite	Messages parlés	Conversations verbales
	Messages écrits	Conversations persistantes
	Messages gestuels	Indication
		Dessin
		Démonstration
Déictique références	Pointage + Conversation	
Manifestation d'actions	Actions stylisées	
Rassemblement d'informations	Conscience de base du groupe	Observation de ceux qui se trouvent dans l'espace de travail, que font-ils et où travaillent-ils
	Informations sur l'activité à partir des objets (Feedthrough)	Changement sur les objets
		Signes ou sons caractéristiques
	Informations sur l'activité à partir des corps des gens (Communication conséquencielle)	Mouvement caractéristique
		Position et localisation du corps
		Direction des regards
Entente surprise des communications explicites des autres (Overhearing)	Présence des autres qui parlent	
	Contenu spécifique qui intéresse les personnes qui l'entendent	
Evidence visuelle	Actions normales	
Accès partagés (aux outils, objets, espaces et temps)	Obtention de ressource	Prendre des objets ou outils physiquement
		Occuper un espace
	Réservation de ressource	Déplacer vers une plus grande proximité
		Notifier les autres de l'intention
	Protection du travail	Surveiller les actions des autres
		Notifier les autres de l'intention
Transfert	Transfert d'objets	Donner/prendre physiquement des objets
		Offrir/accepter verbalement des objets
	Dépôt d'objets	Placer des objets et notifier les autres

### Actions :

Les actions sont des moyens communs pour effectuer une tâche secondaire. Par exemple, si une sous-tâche de collaboration est d'identifier un objet (communication explicite), des actions pour l'accomplissement de cette sous-tâche peuvent être la communication verbale ou le pointage.



Notre modèle de tâche contient trois types d'informations : les informations générales sur la tâche, les informations sur les ressources nécessaires à l'exécution de la tâche, et enfin les informations sur son exécution.

Les informations générales sur la tâche contiennent les informations suivantes :

- Le titre permet d'identifier la tâche.
- Le but représente l'état désiré du système après l'exécution de la tâche.
- La fréquence d'utilisation de la tâche.
- Le type de la tâche (individuelle ou collaborative).

Les informations sur les ressources nécessaires à l'exécution de la tâche décrivent les ressources ainsi que les différentes contraintes concernant celles-ci : les droits d'accès à ces ressources par exemple.

Les informations sur l'exécution de la tâche sont :

- La durée d'exécution de la tâche.
- Les contextes qui décrivent par exemple les contraintes géographiques, les contraintes temporelles, ou d'autres pré-conditions nécessaires à la réalisation de la tâche.
- Le contexte global qui peut être : un contexte créatif et exploratoire, un contexte critique ou un contexte normale.

### 3.3.4 Le modèle des méthodes

Puisque la méthodologie est basée sur la méthode cognitive walkthrough, nous sommes donc amenés à construire notre modèle des méthodes à partir des différentes versions et extensions que nous avons décrites dans le chapitre 2. Notre modèle est décrit dans les paragraphes qui suivent.

Tout d'abord, le nom identifie la méthode ; par exemple "Norton's Cognitive Walkthrough" ou "Extended Cognitive Walkthrough". Tandis que la description donne plus de détails sur la méthode.

Ensuite, le modèle cognitif exploité dans la méthode en question doit être décrit. En effet, on a constaté qu'une méthode est basée sur une ou plusieurs théories cognitives. Par exemple la méthode Norman Cognitive Walkthrough est basée sur la théorie de Norman tandis que la méthode Distributed Cognitive Walkthrough est basée sur la théorie de la cognition distribuée et la théorie de la tâche cognitive distribuée.

L'exécution de la méthode est guidée par des questions auxquelles les évaluateurs doivent répondre. Le type de la question détermine la nature de la réponse à donner. Nous avons quatre types de question :

- Le type numérique : une question de type numérique nécessite une réponse de valeur numérique.
- Le type échelle : une question de type échelle nécessite une réponse appartenant à une liste d'échelons ordonnée. Par exemple, dans la section 2.1 du formulaire 3b de la deuxième

version de CW (Polson et al., 1992), il y a une question de type échelle qui nécessite une réponse parmi la liste suivante : 0, 25, 50, 75, 100.

- Le type nominal : une question de type nominal nécessite une réponse appartenant à un ensemble de valeurs possibles. A la différence de la liste d'échelons des questions de type échelle, l'ensemble de valeur n'est pas ordonné.
- Le type libre : une question de type libre n'impose pas de contrainte à la réponse. L'évaluateur est libre d'exprimer une réponse sémantiquement valable ; c'est-à-dire une réponse liée sémantiquement à la question. Par exemple, la question numéro 3 de la première version de CW est une question de type libre.

On dispose de questions pour les tâches simples et d'autres questions pour les tâches coopératives. Dans notre modèle, nous avons divisé ces questions en quatre catégories :

- La catégorie des questions à répondre avant l'exécution de l'action ; ces questions permettent de préciser ce que pense l'utilisateur avant d'exécuter la tâche.
- La catégorie des questions à répondre pendant l'exécution de l'action.
- La catégorie des questions à répondre après l'exécution de l'action : elles permettent de connaître si l'utilisateur peut comprendre les résultats de l'exécution de l'action et s'il peut continuer l'exécution de l'action suivante pour atteindre son but.
- La catégorie des questions correspondantes aux tâches coopératives.

La méthode qualifiée d'annexe décrit une autre méthode éventuellement associée à la méthode concernée. Outre l'évaluation par la méthode principale, une autre évaluation est dans ce cas effectuée en utilisant la méthode annexe. Cette dernière a le plus souvent l'objectif d'améliorer la découverte des problèmes d'utilisabilité. Par exemple, Heuristic Walkthrough utilise comme méthode annexe une méthode d'évaluation à base d'heuristiques. Il en est de même pour la méthode « Cognitive Walkthrough With Users » qui utilise une méthode de verbalisation à haute voix comme méthode annexe.

La méthode annexe permet d'évaluer l'IHM selon divers aspects. Ainsi, pour Heuristique Walkthrough, on peut utiliser comme méthode annexe, l'évaluation heuristique de Nielsen [Nielsen, 1994], les aspects de l'interface à considérer sont dans ce cas les dix heuristiques de Nielsen [Nielsen, 1994]. A chaque heuristique correspond une liste de questions pour guider l'évaluateur à évaluer l'aspect correspondant. Par exemple, à l'heuristique « Visibilité de l'état du système », Pierotti [2004] propose 29 questions.

Notre modèle permet de transformer certaines questions de type libre en questions nominales ou en questions à échelle. Ces transformations sont utiles pour l'automatisation. Ainsi par exemple, la question Q3 (Comment l'utilisateur accède-t-il à la description de l'action ?) de la première version de CW est transformée en « Est-ce que l'utilisateur peut accéder à la description de l'action ? ». Selon notre modèle, la réponse a pour valeur Oui ou Non et la justification peut répondre à la question dans sa version originale. L'avantage de la transformation est qu'il est facile de détecter les problèmes d'utilisabilité. En effet une valeur de réponse « non » signifie qu'il y a un problème d'accessibilité lié à la description de l'action, tandis que dans sa version originale il est difficile de déduire s'il y a un problème d'utilisabilité.

Avec ce modèle on a pu représenter dans l'état actuel des travaux deux versions et six variantes de CW ainsi qu'une méthode annexe. Les détails de ces représentations sont décrits ci-

après. Dans cette liste on a une méthode à deux passes à savoir Heuristic Walkthrough. Cette méthode utilise comme méthode principale la troisième version de CW (CW3) et comme méthode annexe la méthode heuristique de Nielsen.

### **3.3.4.1 Représentation de la première version de Lewis et al. [1990] (CW1)**

**Nom :**

CW1

**Description :**

CW1 est la première version de CW. Elle ne nécessite pas beaucoup de ressource. Un investissement relativement limité en ressources, environ une heure par tâche par interface, permet de détecter près de 50 pour cent des problèmes rencontrés par les utilisateurs.

**Modèle cognitif :**

Modèle de l'apprentissage par exploration (CE+)

Pour cette méthode sui contient des questions de type libre, nous avons fait des transformations. La référence à la fin de chaque question correspond au numéro de la question selon la description dans [Lewis et al., 1990].

**Questions avant action :**

Q1 : Est-ce évident que l'action est disponible ? (2a)

Q2 : Est-ce évident que l'action est appropriée au but ? (2b)

Q3 : Est-ce que l'utilisateur peut accéder à la description de l'action ? (3 transformée)

Q4 : Est-ce que l'utilisateur peut associer la description à l'action ? (4 transformée)

Q5 : Est-ce que les autres actions disponibles sont moins appropriées ? (5)

Q6 : Est-ce que l'utilisateur peut exécuter l'action ? (6 transformée)

**Questions pendant l'action :**

Q7 : Est-ce que l'utilisateur peut prendre sa décision à temps dû au délai d'attente de l'interface ? (7 transformée)

**Questions après action :**

Q8 : Est-il évident qu'on progresse vers le but ? (8a)

Q9 : Est-ce que l'utilisateur peut accéder aux informations nécessaires contenues dans la réponse du système ? (8b)

Q10 : Si le but a changé, est-ce évident qu'il a changé ? (9a)

Q11 : Si la tâche est terminée, est ce évident qu'elle est terminée ? (9b)

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune.

### **3.3.4.2 Représentation de la troisième version de Warton et al. [1994] (CW3)**

**Nom :**

CW3

**Description :**

CW3 est la troisième version de la méthode cognitive walkthrough. C'est une version suffisamment souple pour s'adapter dans le processus de développement de logiciel. La méthode permet d'identifier les problèmes au début du processus et de décrire les raisons de ces problèmes.

**Modèle cognitif :**

Modèle de l'apprentissage par exploration (CE+)

**Questions avant action :**

- Q1 : Est-ce que l'utilisateur tente de réaliser l'effet désiré ?
- Q2 : Est-ce que l'utilisateur constate que l'action correcte est disponible ?
- Q3 : Est-ce que l'utilisateur associe l'action correcte avec l'effet qu'il tente de réaliser ?

**Questions pendant action :**

Aucune

**Questions après action :**

- Q4 : Si l'action correcte est exécutée, est-ce que l'utilisateur se rend compte qu'il progresse vers la réalisation de la tâche ?

**Questions pour les tâches coopératives :**

Aucune

**Méthode annexe :**

Aucune

### **3.3.4.3 Représentation de Heuristic Walkthrough (HW)**

**Nom :**

Heuristic Walkthrough

**Description :**

Méthode d'évaluation à deux passes combinant une méthode CW et une méthode d'évaluation à base d'heuristique. La version originale de Sears combine la troisième version (CW3) et l'évaluation heuristique de Nielsen.

**Modèle cognitif :**

Modèle de l'apprentissage par exploration (CE+)

**Méthode principale :**

CW3

**Méthode annexe :**

Toutes les méthodes à base d'heuristiques peuvent être utilisées. En particulier l'évaluation heuristique de Nielsen (Nielsen, 1994), la méthode ISO 9241 Evaluator (Reitterer et Oppermann, 1995 ; Oppermann et Reitterer, 1997), la méthode IsoMetrics (Gediga, Hamborg, et Düntsch, 1999).

### 3.3.4.4 Représentation de Norton CW

**Nom :**

Norton CW

**Description :**

Norton CW est une version de CW basée sur le modèle de Norman. Elle permet de faire face aux problèmes découlant de la distribution des équipes collaborant au projet dans plusieurs villes (parfois différents pays), et la nécessité d'affronter les problèmes de haut niveau d'interaction dans le processus d'évaluation. C'est intéressant pour une première approche d'évaluation d'un système interactif, sans entrer dans les détails de sa réalisation.

**Modèle cognitif :**

Modèle de l'action de Norman

**Questions avant action :**

- Q1 : Est-ce que l'action correcte et faisable est suffisamment mise en évidence pour l'utilisateur et est-ce que les actions correspondent avec l'intention exprimée par l'utilisateur ?
- Q2 : Est-ce que l'utilisateur relie la description de l'action correcte avec ce qu'il essaie de faire ?

**Questions pendant action :**

Aucune.

**Questions après action :**

- Q3 : Est-ce que l'utilisateur reçoit le retour d'information au même endroit et avec la même modalité suite à l'exécution de l'action ?
- Q4 : Est-ce que l'utilisateur interprète correctement la réponse du système sur l'action choisie, c'est-à-dire est-ce qu'il réalise qu'il a fait le bon ou mauvais choix ?
- Q5 : Est-ce que l'utilisateur évalue correctement les résultats, c'est-à-dire qu'il est capable d'évaluer s'il s'approche du but ?

Q6 : Si le but n'est pas bon (ou peut être amélioré), est-ce qu'il comprend que l'intention qu'il essaie d'exécuter ne peut pas être accomplie dans l'état actuel (ou est-ce qu'il trouve des buts alternatifs pour l'améliorer) ?

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune

### 3.3.4.5 Représentation de Streamlined CW

**Nom :**

Streamlined CW

**Description :**

Streamlined CW variante apporte une dimension sociale dans l'évaluation. Les membres de l'équipe de développement sont contraints par le temps, tendent à tomber dans de longues discussions de conception, et sont parfois sur la défensive relativement à leur conception. Cette méthode permet de surmonter ces contraintes sociales.

**Modèle cognitif :**

Modèle de l'apprentissage par exploration (CE+)

**Questions avant action :**

Q1 : Est-ce que l'utilisateur sait ce qu'il doit faire à cette étape ?

**Questions pendant action :**

Aucune.

**Questions après action :**

Q2 : Si l'utilisateur réalise correctement l'action, est-ce qu'il saura qu'il a réalisé la bonne action et qu'il progresse vers le but en question ?

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune.

### 3.3.4.6 Représentation de CWW

**Nom :**

CWW

**Description :**

CWW est une version de CW basée sur un modèle de l'exploration basée sur la compréhension. Elle permet d'évaluer dans quelle mesure les sites Web soutiennent les utilisateurs dans leurs tâches de navigation et de recherche d'information

**Modèle cognitif :**

Modèle CoLiDeS (Comprehension-based Linked model of Deliberate Search)

**Questions avant action :**

- Q1 : Est-ce que l'action correcte est suffisamment mise en évidence pour l'utilisateur ?
- Q2 : Est-ce que l'utilisateur relie la zone correcte de la page avec le but et ceci en utilisant l'information sur le titre et la compréhension de la convention de structuration de la page ?
- Q3 : Est-ce que l'utilisateur relie le but avec les outils graphiques corrects dans la zone en question et ceci en utilisant des étiquettes des liens et d'autres informations descriptives ?

**Questions pendant action :**

Aucune.

**Questions après action :**

- Q4 : Est-ce que l'utilisateur interprète correctement la réponse du système avec l'action choisie ?

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune.

### 3.3.4.7 Représentation de Extended CW

**Nom :**

Extended CW

**Description :**

Extended CW est une méthode intéressante pour les évaluateurs possédant peu de connaissance en psychologie cognitive. Elle est prévue en outre pour détecter plus facilement les problèmes d'accessibilité.

**Modèle cognitif :**

Modèle de l'action de Norman.

**Questions avant action :**

- Q1 : Est ce que l'utilisateur a l'intention de réaliser l'action correcte ?

Q2 : Est-ce que l'utilisateur s'aperçoit que l'objet correct est disponible ?

Q3 : Est-ce que l'utilisateur sait sur quoi l'action correcte se rapporte ?

Q4 : Est-ce que l'utilisateur s'aperçoit que l'action correcte est disponible ?

Q5 : Est-ce que l'utilisateur sait que l'action correcte doit être appliquée à l'objet en question ?

Q6 : Est-ce que l'utilisateur est capable d'appliquer l'action correcte sur l'objet concerné sans erreur ou difficulté ?

**Questions pendant action :**

Aucune.

**Questions après action :**

Q7 : Quand l'action correcte est prise, est-ce que l'utilisateur aperçoit le changement sur l'état du système ?

Q8 : Est-ce que l'utilisateur sait ce qui se passait concernant le changement d'état du système ?

Q9 : Est-ce que l'utilisateur sait que l'état courant du système s'approche de l'achèvement de la tâche ?

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune.

### 3.3.4.8 Représentation de Enhanced CW

**Nom :**

Enhanced CW

**Description :**

Enhanced CW est une variante qui emploie une procédure clairement détaillée pour la simulation de processus de résolution de problème de l'utilisateur à chaque étape de l'interaction. Tout au long de l'interaction, il faut vérifier si l'objectif fixé du futur utilisateur ainsi que son expérience conduira à la prochaine action correcte.

**Modèle cognitif :**

Modèle de l'apprentissage par exploration (CE+)

**Questions avant action :**

Q1 : Est-ce que l'utilisateur sait que la fonction évaluée est disponible ?

Q2 : Est-ce que l'utilisateur essaye d'atteindre le bon effet ?

Q3 : Est-ce que l'interface utilisateur donne des indices qui montrent que la fonction est disponible ?



Q4 : Est-ce que l'utilisateur est en mesure de s'apercevoir que l'action correcte est disponible ?

Q5 : Est-ce que l'utilisateur associe l'indice adéquat avec la fonction souhaitée ?

**Questions pendant action :**

Aucune.

**Questions après action :**

Q6 : Est-ce que l'utilisateur associe l'action correcte avec l'effet désiré ?

Q7 : Est-ce que l'utilisateur obtient un retour d'information suffisant pour comprendre que la fonction désirée a été choisie ?

Q8 : Si la bonne action est effectuée, est-ce que l'utilisateur voit que des progrès sont réalisés vers la solution de la tâche ?

Q9 : Est-ce que l'utilisateur obtient un retour d'information suffisant pour comprendre que la fonction désirée a été effectuée ?

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune.

### 3.3.4.9 Représentation de Distributed CW

**Nom :**

Distributed CW

**Description :**

DCW est une variante qui offre des perspectives intéressantes dans le cadre des interactions en général, c'est-à-dire pas seulement entre un humain et un système interactif. Elle est basée sur la cognition distribuée.

**Modèle cognitif :**

Modèle de la cognition distribué (Hollan et al., 2000) et modèle de la tâche cognitive distribuée (Zhang et Norman, 1994).

**Questions avant action :**

Q1 : Est-ce que la façon dont l'information est représentée fournit toutes les connaissances nécessaires pour mener à bien la tâche ?

Q2 : Est-ce que la façon dont l'information est présentée montre les progrès antérieurs en rapport avec la mission générale ?

Q3 : Est-ce que la façon dont l'information est représentée fournit des ressources évitant à l'utilisateur d'avoir à réfléchir ou de calculer quoi que ce soit en menant à bien sa tâche ?

**Questions pendant action :**

Aucune.

**Questions après action :**

Q4 : Si la tâche en cours est réalisée, est-ce que la manière dont l'information est représentée a été modifiée d'une manière que le résultat de la tâche est accessible par les utilisateurs actuels ou d'autres utilisateurs à un moment ultérieur ou à un endroit différent ?

**Questions pour les tâches coopératives :**

Aucune.

**Méthode annexe :**

Aucune.

### 3.3.4.10 Représentation de la méthode d'évaluation heuristique de Nielsen

**Nom :**

Evaluation Heuristique de Nielsen

**Description :**

C'est une méthode d'évaluation basée sur la liste d'heuristiques d'utilisabilité de Nielsen.

**Modèle cognitif :**

Aucun.

**Questions avant action :**

Aucune.

**Questions pendant action :**

Aucune.

**Questions après action :**

Aucune.

**Questions pour les tâches coopératives :**

Aucune.

**Questions annexes :**

- Visibilité de l'état du système (environ 29 questions)
- Relation entre le système et le monde réel (environ 24 questions)
- Contrôle de l'utilisation et la liberté (environ 23 questions)
- Cohérence et normes (environ 51 questions)

- Prévention d'erreurs (environ 21 questions)
- Reconnaissance plutôt que de rappeler (environ 40 questions)
- La flexibilité et l'efficacité d'utilisation (environ 16 questions)
- Esthétique et design minimaliste (environ 12 questions)
- Aidez les utilisateurs à reconnaître, diagnostiquer et récupérer des erreurs (environ 21 questions)
- Aide et documentation (environ 23 questions)

### 3.3.4.11 Représentation de Groupware Walkthrough (GW)

**Nom :**

GW

**Description :**

La méthode GW est une modification profonde de CW visant à inclure des facteurs de complexité liés au fonctionnement d'une équipe de travail.

**Modèle cognitif :**

Aucun.

**Questions avant action :**

Aucune.

**Questions pendant action :**

Aucune.

**Questions après action :**

Après l'exécution de l'action, est-ce qu'il y a des problèmes ?

**Questions pour les tâches coopératives :**

Q1 : Est-ce que la tâche peut être exécutée efficacement, est-ce que l'interface fournit le moyen de le faire et correctement ?

Q2 : Est-ce que la tâche peut être effectuée de façon efficiente ?

Q3 : Est-ce que la tâche est exécutée avec satisfaction ?

**Méthode annexe :**

Aucune.

## 3.3.5 Le modèle de résultat

Les résultats du processus d'évaluation comprendront les traces de l'évaluation et les problèmes d'utilisabilité trouvés. Les traces de l'évaluation enregistrent les réponses des évaluateurs lors de l'inspection. C'est la dématérialisation des fiches d'évaluation ou du rapport de description des problèmes.

Les problèmes d'utilisabilité comprendront les informations suivantes :

- Numéro du problème : il s'agit d'un numéro correspondant à l'ordre d'apparition du problème.
- Référence du problème : il s'agit de la référence de l'action origine du problème. Ce qui permettra aux lecteurs du rapport de connaître l'origine de celui-ci.
- Découverte du problème : il s'agit du moyen de découverte du problème. Il faut donner le numéro de la question.
- Description du problème : Selon [Lavery, 1997], la description du problème peut être obtenue en reformulant la question qui a permis de découvrir ce problème. Ensuite il faut décrire aussi les conséquences du problème.
- Pourcentage des utilisateurs qui peuvent rencontrer le problème : les évaluateurs sont invités à estimer le pourcentage des utilisateurs qui peuvent rencontrer ce problème en se basant sur le modèle de l'utilisateur.
- Fréquence du problème : les évaluateurs sont invités à estimer la fréquence du problème (Rarement, Souvent, Très souvent, Toujours).
- Gravité du problème : les évaluateurs sont invités à estimer la gravité du problème. Un problème est « Tolérable » si on peut l'ignorer même si la gêne a été soulignée. Il est modéré s'il provoque une gêne ou une perte de temps aux utilisateurs. Il est sérieux s'il dégrade le contrat de service. Et enfin, il est critique s'il interrompt l'exécution de la tâche.
- Catégorie du problème : il s'agit de généraliser la proposition de Bligard et Osvalder [2007], afin de faciliter le traitement des résultats de l'évaluation. Nous proposons donc d'adopter les six catégories de problèmes de Bligard et Osvalder [2007] : les problèmes liés à l'expérience et la connaissance de l'utilisateur ; les problèmes liés à l'affichage de l'information, les problèmes liés à l'illustration de l'information, les problèmes liés à la séquence d'actions, les problèmes liés au retour d'information et les problèmes qui ne sont pas classables dans les cinq catégories précédentes.
- Solution du problème : les évaluateurs sont invités à proposer une ou des solutions pour palier le problème.

Notons que la structure du rapport de description de problèmes qui vient d'être décrite est conforme aux trois premières recommandations de Jeffries [1994] : descriptions séparées des problèmes et solutions, fourniture des justifications des problèmes et estimation des gravités des problèmes.

## 3.4 Modélisation dynamique de l'évaluation

Dans cette section, nous allons décrire (spécifier) les interactions entre les différentes entités à chaque étape de l'évaluation ainsi que les interactions entre les acteurs de l'évaluation et l'environnement CWE.

### 3.4.1 Spécification de l'artefact

Nous allons décrire le processus de spécification de l'artefact sur la base du modèle de l'artefact (cf. §3.3.1). Ce processus met en jeu entre autre les classes « Artefact », « Domaine »,

« Rôle », « Fonctionnalité », « Lexique », etc (cf. figure 3.2). On va voir les deux activités de ce processus : description du système et constitution des lexiques.

### 3.4.1.1 Description du système

« Décrire le système à évaluer » consiste à fournir les informations concernant l'application. Elle permet de saisir ou modifier les informations suivantes :

- Le nom et la description de l'application,
- Le domaine de l'application,
- Le contexte d'utilisation et la motivation de l'application,
- les différents rôle que les utilisateurs peuvent jouer dans le système à évaluer.
- Les fonctionnalités du système que les utilisateurs peuvent utiliser pour exécuter une tâche. Une fonctionnalité correspond à une partie de l'application tandis qu'une opération est le moyen d'utiliser la fonctionnalité [Jacobson & John, 2000]. Par exemple la fonctionnalité « édition » de l'application *Word* possède les opérations suivantes : copier, coller, couper, etc. Pour l'environnement CWE, on ne décrit pas les opérations.

La tâche description du système est décrite par le diagramme de la figure 3.3.

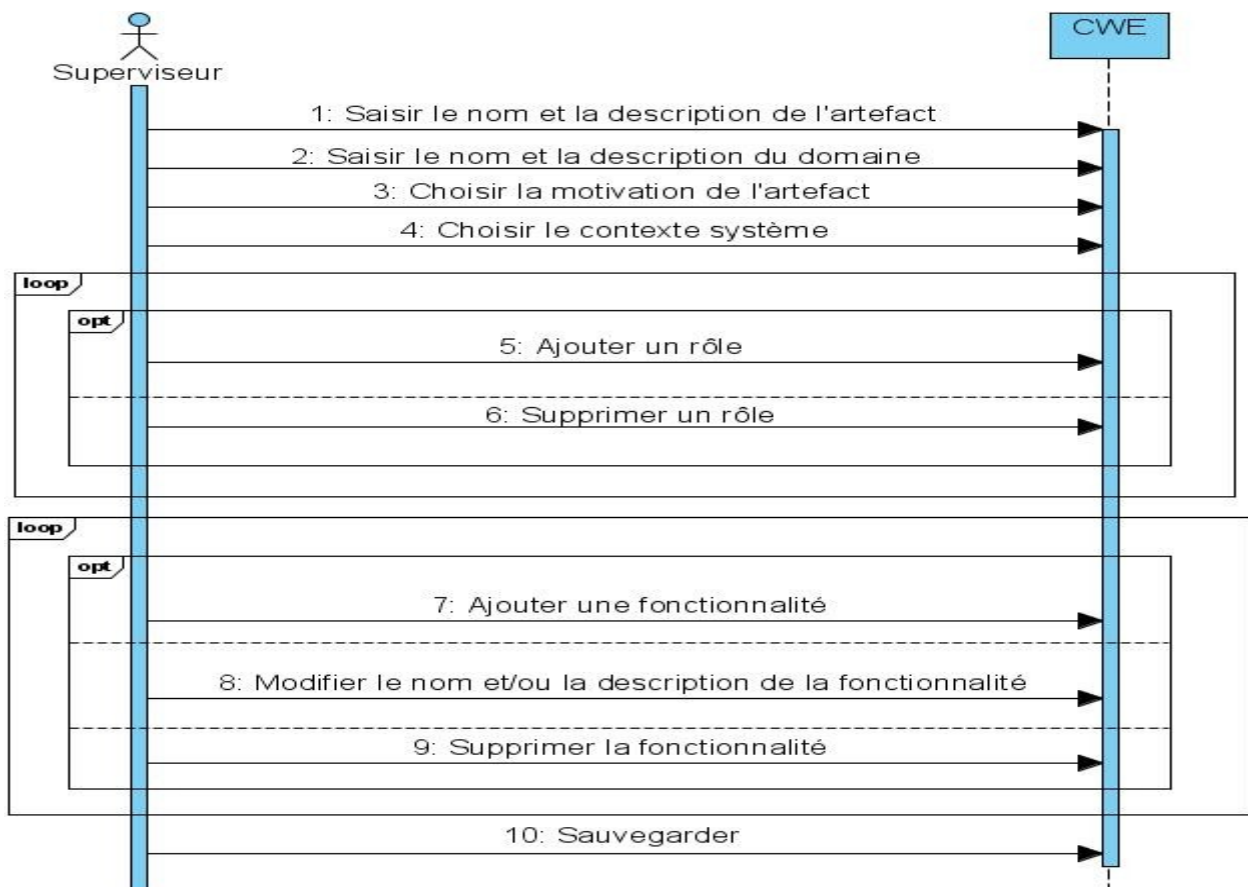


Figure 3.3 : Spécification du système à évaluer

### 3.4.1.2 Constitution des lexiques

Puisque les lexiques permettent aux évaluateurs de bien connaître les concepts du domaine, nous proposons d'utiliser la méthode décrite par Leite & Franco [1993] et Leite et al. [2000] pour les constituer.

Selon cette méthode, la construction des lexiques est basée sur l'idée d'acquisition de vocabulaire. L'acquisition de vocabulaire consiste en la compréhension du langage du problème sans se soucier de la compréhension du problème.

Il existe quatre catégories de lexique (voir tableau 3.2) à savoir le « sujet », le « verbe », l'« objet » et l'« état ». Chaque lexique est décrit par des « notions » et des « comportements ». Une « notion » (dénotation) décrit le lexique (QUOI) et le « comportement » (connotation) permet d'obtenir des informations sur le contexte (COMMENT, OU, QUAND, etc.).

Les descriptions des « notions » et « Comportements » doivent respecter deux principes. D'abord le principe de circularité qui tente de maximiser l'utilisation des lexiques du domaine pour la description des autres lexiques. Ensuite le principe de « vocabulaire minimal » qui tente de minimiser l'utilisation d'autres mots n'appartenant aux lexiques du domaine.

La constitution des lexiques est décrite dans le diagramme de la figure 3.4.

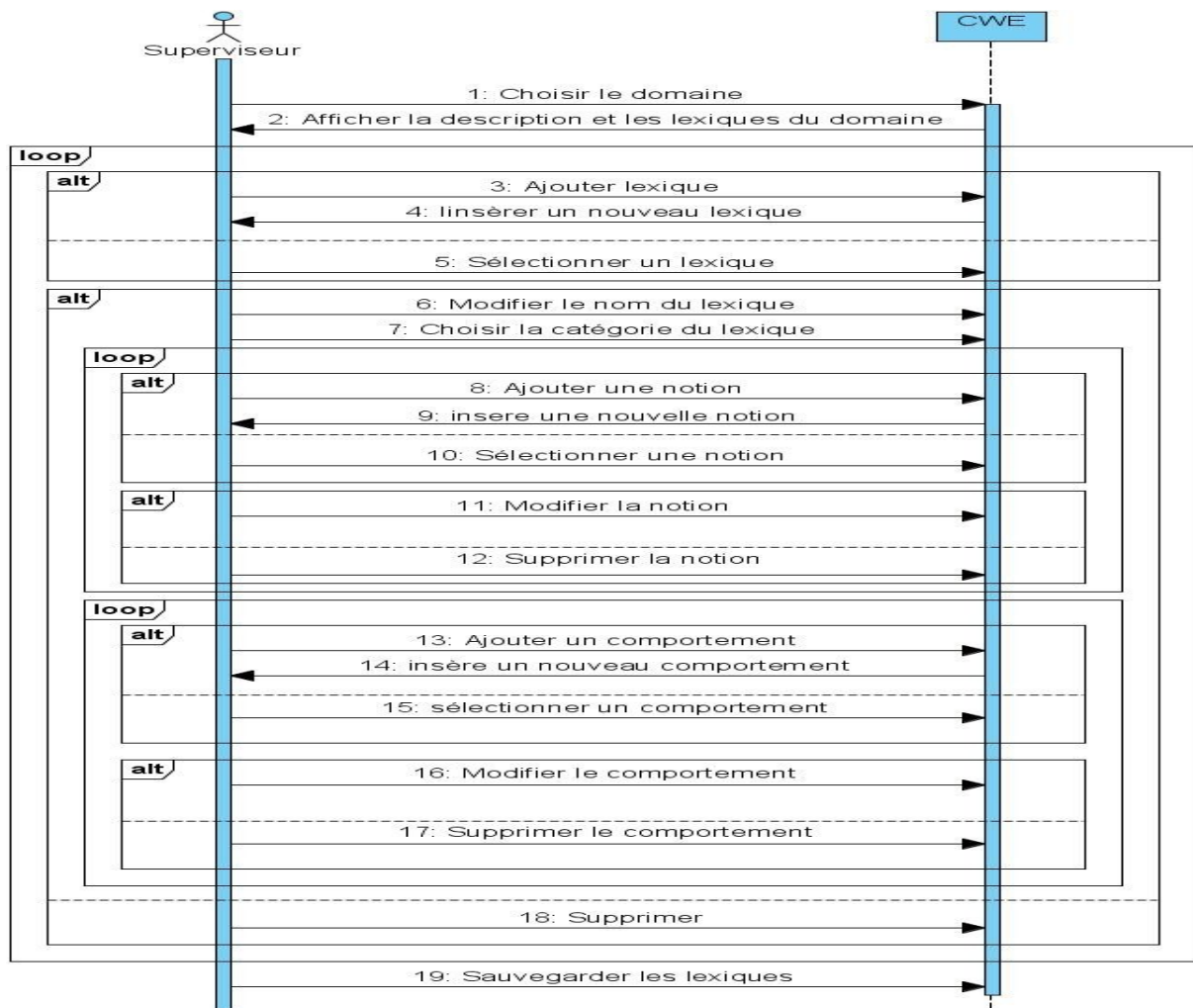


Figure 3.4 : Constitution des lexiques

**Tableau 3.2 : Les quatre catégories de lexiques**

Catégories	Notion	Comportement
Sujet	Qui es le sujet ?	Quelles actions exécute le sujet ?
Verbe	Qui font ?, quand fait-on ?, Comment fait-on ?	Quelles sont les conséquences sur l'environnement ? Quel sont les nouveaux états ?
Objet	Qu'est-ce que c'est ?	Quelles sont les actions qu'on peut appliquer à l'objet ?
Etat	Qu'est-ce que ça veut dire ? Et quelles actions peuvent conduire à cet état ?	Quelles autres

## 3.4.2 Modélisation de l'utilisateur

Nous allons décrire le processus de modélisation des utilisateurs du système à évaluer sur la base du modèle de l'utilisateur (cf. §3.3.2). Ce processus met en jeu entre autre les classes utilisateur, artefact, domaine, fonctionnalité, .etc (cf. figure 3.2). On spécifie pour un système précédemment spécifié selon le processus de spécification du système à évaluer (cf. 3.4.1), l'identité de l'utilisateur, ses connaissances en informatique et ses expériences concernant le système ainsi que ses différentes fonctionnalités.

### 3.4.2.1 Identification de l'utilisateur

On a vu dans le chapitre précédent que la modélisation de l'utilisateur peut être vue comme la collecte et l'analyse des informations sur les utilisateurs. Ces informations appartiennent à deux catégories [Robert, 2003] : les informations génériques et indépendantes de l'application et les informations spécifiques et dépendantes de l'application. Les informations de la première catégorie peuvent être obtenues à partir de certaines études scientifiques, statistiques, démographiques, sociologiques ou de différents sondages et enquêtes ainsi que d'études de marché. En revanche, les informations de la deuxième catégorie peuvent être obtenues, soit explicitement à partir des interviews, soit implicitement à partir des captures de données et observations basées sur l'interaction des utilisateurs avec le système [Razmerita, 2003].

La tâche « Identification de l'utilisateur » est décrite par le diagramme de la figure 3.5.

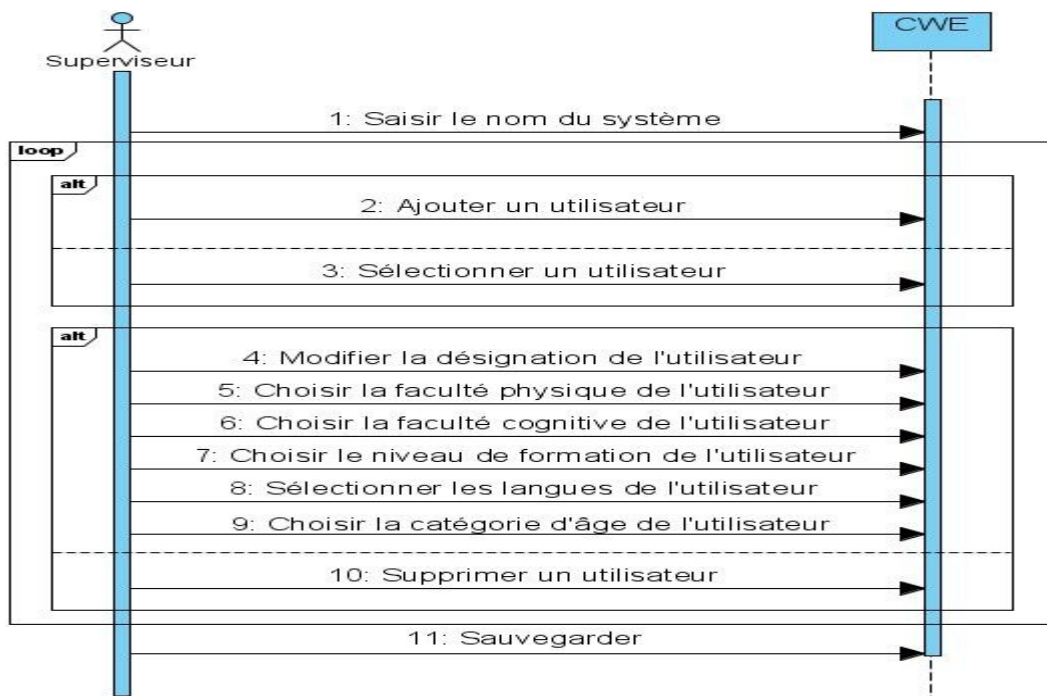


Figure 3.5 : Identification de l'utilisateur

### 3.4.2.2 Description de la connaissance en informatique de l'utilisateur

La description de la connaissance en informatique de l'utilisateur consiste à choisir entre « faible », « Moyenne » ou « Bonne » pour chaque plateforme (PC, Mac, Station de travail, Mobile) et pour chaque système (Unix Window, MacOS). Cette tâche est décrite par le diagramme de la figure 3.6.

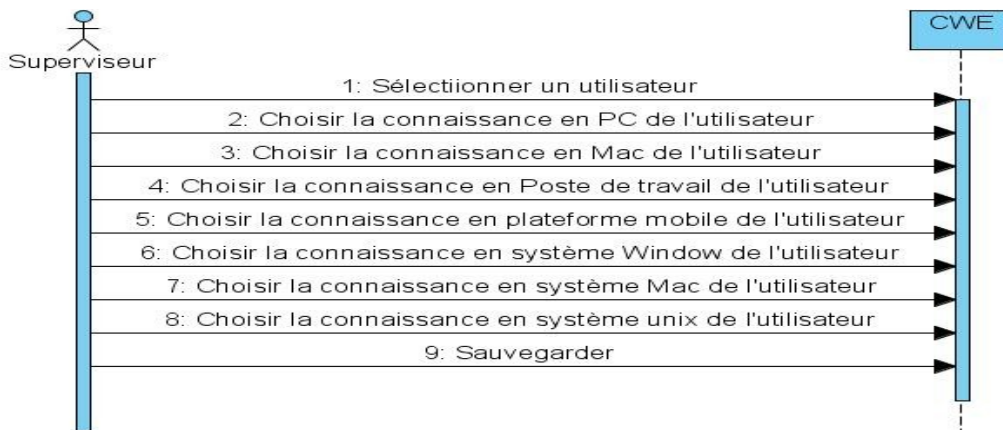


Figure 3.6 : Description de la connaissance en informatique de l'utilisateur

### 3.4.2.3 Description de la connaissance de l'utilisateur sur l'artefact

La description de la connaissance de l'utilisateur sur l'artefact consiste à fournir des informations sur les connaissances et expériences (connaissance, expérience, fréquence d'utilisation, type d'utilisation et motivation) de l'utilisateur concernant l'artefact dans sa globalité et éventuellement de façon détaillée (par fonctionnalité et/ou opération). Cette tâche est décrite par le diagramme de la figure 3.7.



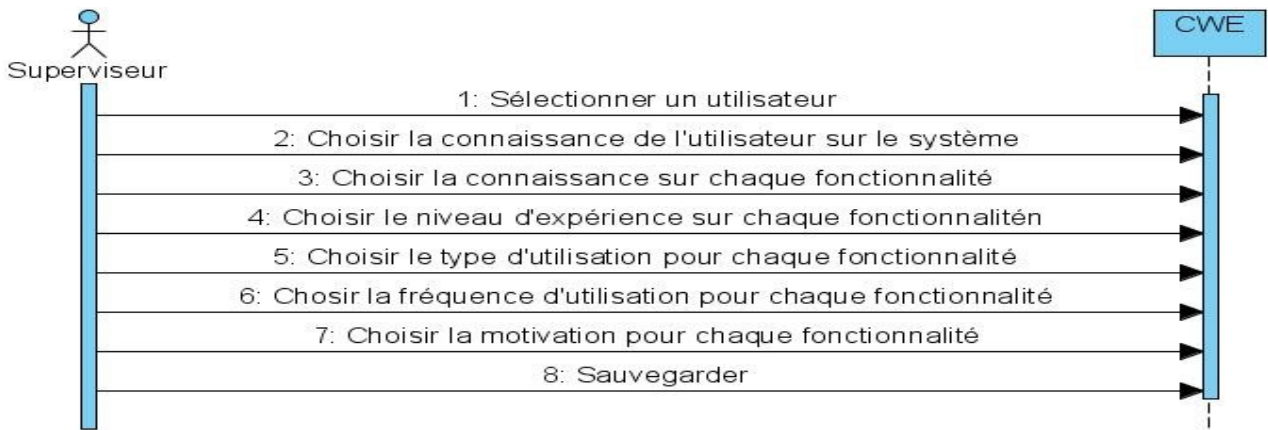


Figure 3.7 : Description de la connaissance de l'utilisateur sur l'artefact

### 3.4.3 Choix de scénarii à évaluer

Le déroulement de cette tâche est décrit dans la figure 3.8. Elle consiste à décrire de scénarii considérés comme réalistes et importants pour un système précédemment spécifié. Nous allons spécifier ce qu'est un tel scénario.

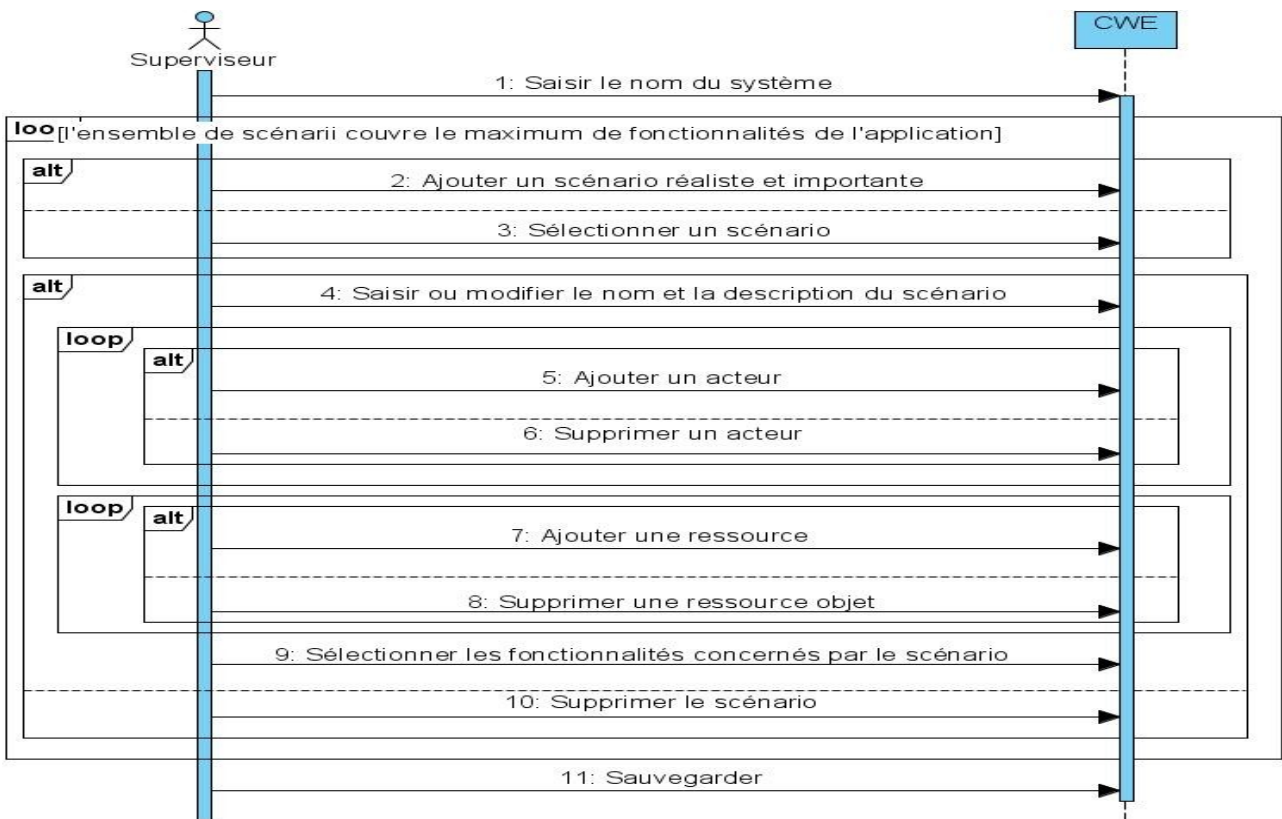


Figure 3.8 : Choix de scénarii à évaluer

La méthode CW ne fournit pas de guide pour choisir les tâches à évaluer parmi plusieurs tâches supportées par l'application. La raison est que le modèle théorique ou modèle cognitif de la méthode ne couvre pas le choix de tâches. D'un autre côté, on ne peut pas évaluer toutes les tâches supportées par l'application à cause du coût. Néanmoins, Wharton et al. [1992] et Lewis & Wharton

[1997] proposent deux critères pour appuyer le choix de tâches à évaluer : les tâches doivent être importantes, elles doivent aussi être réalistes.

Une tâche importante est une tâche qui concerne le cœur de l'application. C'est une tâche qui va être utilisée le plus souvent. Elle peut être aussi de nature critique.

Une tâche réaliste est une tâche qui reflète le mieux la nature réelle des travaux des utilisateurs.

En ce qui concerne le nombre de tâches, Jacobsen & John [2000] proposent de choisir quelques scénarios qui couvrent l'ensemble des fonctionnalités de l'application. Et pour connaître la couverture de l'ensemble des fonctionnalités de l'application de chaque scénario, il suffit de remplir le tableau de couverture de scénarii (voir tableau 3.3).

Nous proposons donc de choisir quelques scénarios parmi les plus réalistes et les plus importants d'après les deux critères cités ci-dessus. Ces scénarios donneront ensuite une liste de tâches à évaluer. Le superviseur peut vérifier la couverture des tâches et peut en ajouter d'autres jusqu'à ce que l'ensemble des tâches choisies couvre la totalité des fonctionnalités de l'application.

**Tableau 3.3 : Tableau de couverture de scénarii**

Fonctionnalités	Scénario 1	Scénario 2	.....	Scénario n
Fonctionnalité 1	X			X
Fonctionnalité 2		X		X
.....				
Fonctionnalité n			X	X
Couverture = (Nombre opérations concernées/Nombre opérations total)*100				

### 3.4.4 Analyse de tâches

L'objectif de l'analyse de tâches est de décrire et décomposer le scénario en un ensemble des tâches, de sous-tâches et des actions (voir figure 3.9).

Une fois le scénario enregistré, les tâches devraient être extraites de la description de l'activité du scénario (une description de l'activité contient généralement plusieurs tâches). L'approche utilisée ici n'est pas de déterminer la séquence absolue des tâches qui se produisent dans le scénario, mais d'identifier les activités qui devraient être soutenus afin de couvrir un éventail raisonnable d'alternatives possibles.

Chaque tâche est ensuite décomposée en sous-tâches en tenant compte de la façon dont la tâche sera exécutée dans le cadre de travail. Pour les sous-tâches collaboratives, le principe de collaboration à utiliser doit être identifié. Les sous-tâches individuelles peuvent être décomposées en suite d'actions en utilisant une méthode d'analyse de tâche classique. Nous proposons l'analyse hiérarchique de tâche (HTA pour Hierarchical Task Analysis) de Annett et Duncan [1967]. Ce choix est motivé par la généricité et la popularité de cette méthode. En effet, HTA est une méthode qu'on peut utiliser dans divers domaines d'application en particulier l'IHM. Kirwan et Ainsworth [1992] affirment que c'est la méthode d'analyse de tâche la plus connue ; en plus, c'est une méthode qui évolue : 30 ans après son premier développement, elle est encore d'actualité et fait l'objet de plusieurs publications scientifiques dont celle de Stanton [2006] qui dresse une revue des développements et extensions de la méthode.

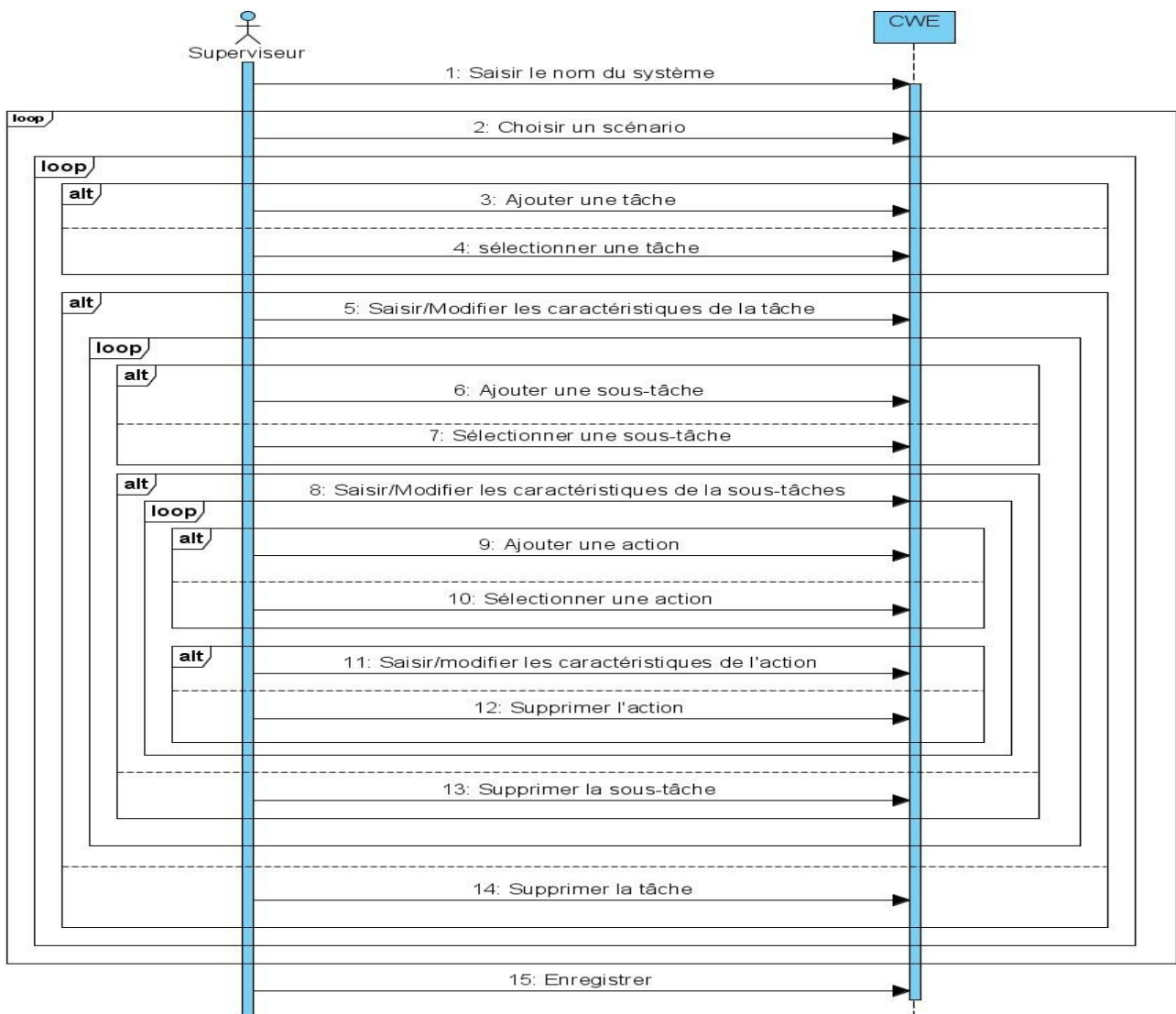


Figure 3.9 : Analyse de tâches

### 3.4.5 Choix de méthode d'évaluation

Nous proposons trois méthodes de choix de méthode d'évaluation :

- La première permet au superviseur le libre choix de la méthode d'évaluation à utiliser.
- La deuxième méthode permet au superviseur de choisir la méthode à utiliser dans une liste restreinte de méthodes que CWE propose.
- Et enfin, la troisième méthode consiste à laisser CWE choisir la méthode à utiliser.

Pour les deux dernières méthodes les critères de choix considérés ici sont des critères purement techniques. Dans l'état actuel des travaux, nous ne considérerons pas les critères liés à la gestion du projet tels que le temps ou le budget. Les trois critères principaux de sélection de méthodes sont le type de la tâche, la motivation et le contexte du système.

Pour une tâche de type collaborative, la méthode proposée est la méthode GW.

La motivation de l'application permet de prioriser les critères ou attributs d'utilisabilité. Pour une tâche à accomplir relativement à une application domestique, comme nous l'avons précisé précédemment, notre hypothèse de travail est qu'il faut considérer en priorité la facilité d'apprentissage et l'esthétique ; tandis que pour une tâche à exécuter sur une application industrielle ou commerciale, la productivité est à considérer en priorité.

Pour les applications domestiques, il faut une méthode qui permette d'évaluer la facilité d'apprentissage et l'esthétique de l'interface. Une méthode basée sur le modèle CE+ semble donc le mieux adapté pour eux. En effet, le modèle CE+ a été prévu pour la conception et l'évaluation des applications qu'on utilisera sans ou avec peu de formation initiale (walk up and use application). Ainsi, la méthode HW est à proposer. La méthode annexe de cette méthode (évaluation heuristique) permet d'évaluer l'esthétique. En plus, les autres méthodes basées sur le modèle CE+ telles que CW1, CW3 sont aussi proposées.

Pour les applications industrielles et commerciales qui exigent des coûts et performances optimisés, aucune méthode parmi celles visées pour l'instant n'est particulièrement destinée à ce type d'application. Donc on peut choisir n'importe quelle méthode de la liste.

Le contexte global de la tâche servira aussi de guide pour choisir la méthode d'évaluation.

Ainsi, une tâche créative et exploratoire oblige ses utilisateurs à explorer plusieurs solutions. Donc, les méthodes basées sur le modèle CoLiDeS qui sont des modèles basés par l'exploration guidée par des objectifs peuvent convenir avec les applications créatives.

Les tâches qu'on pourrait spécifier de normales peuvent apparemment être évaluées avec n'importe quelle méthode de la liste. Aucune spécification n'est trouvée.

Et enfin, en ce qui concerne les applications utilisées dans le contexte des situations, seule la première version possède une question qui s'occupe de la contrainte de temps. Donc dans l'état actuel des travaux nous proposons d'utiliser cette méthode pour ce type d'application.

### 3.4.6 Prise en main de l'évaluation

La prise en main de l'évaluation consiste en l'appropriation des entrées de l'évaluation ainsi que de l'environnement d'évaluation (CWE) et des méthodes d'évaluation par les inspecteurs. Les entrées de l'évaluation sont constituées de la description du système à évaluer préalablement décrite par le superviseur ainsi que les lexiques du domaine concerné par le système.

Pour permettre aux inspecteurs d'apprendre CWE, la documentation de CWE est mise à leur disposition pour qu'ils puissent l'explorer (voir figure 3.11). Il en est de même pour les documentations des différentes versions et variantes de CW (voir figure 3.10).

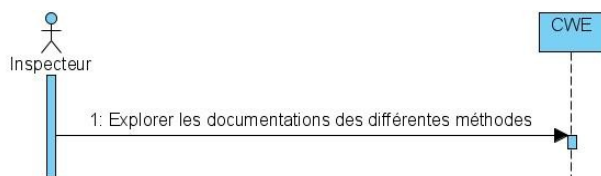


Figure 3.10 : Apprentissage des différentes méthodes

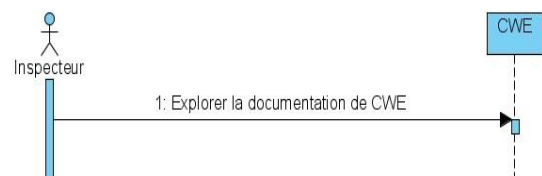


Figure 3.11 : Apprentissage de CWE

Enfin, pour permettre aux inspecteurs de s'approprier les entrées de l'évaluation, CWE leur met à disposition un outil dont la fonctionnalité est décrite par la figure 3.12.

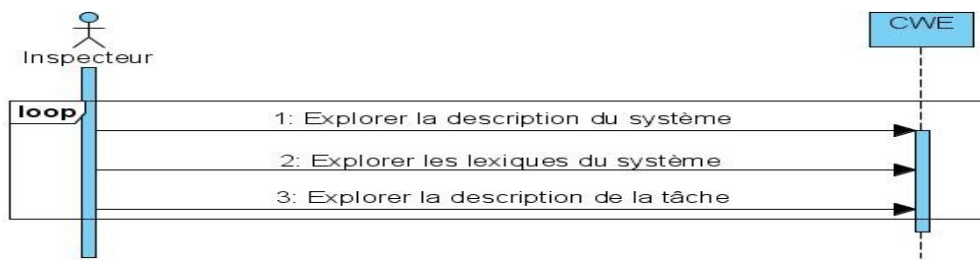


Figure 3.12 : Appropriation des entrées de l'évaluation

### 3.4.7 Identification de l'évaluateur

Chaque évaluateur voulant utiliser l'outil CWE doit s'identifier au début de chaque session sur CWE. Il doit fournir diverses informations en tant qu'utilisateur de CWE. Ces informations doivent donc être conformes au modèle de l'utilisateur. Les fonctionnalités de CWE sont « la spécification du système à évaluer », « la modélisation de l'utilisateur », « le choix de scenarii à évaluer », « l'analyse des tâches », « le choix de méthode d'évaluation », « la prise en main de l'évaluation », « l'inspection des tâches » et « l'analyse des résultats ».

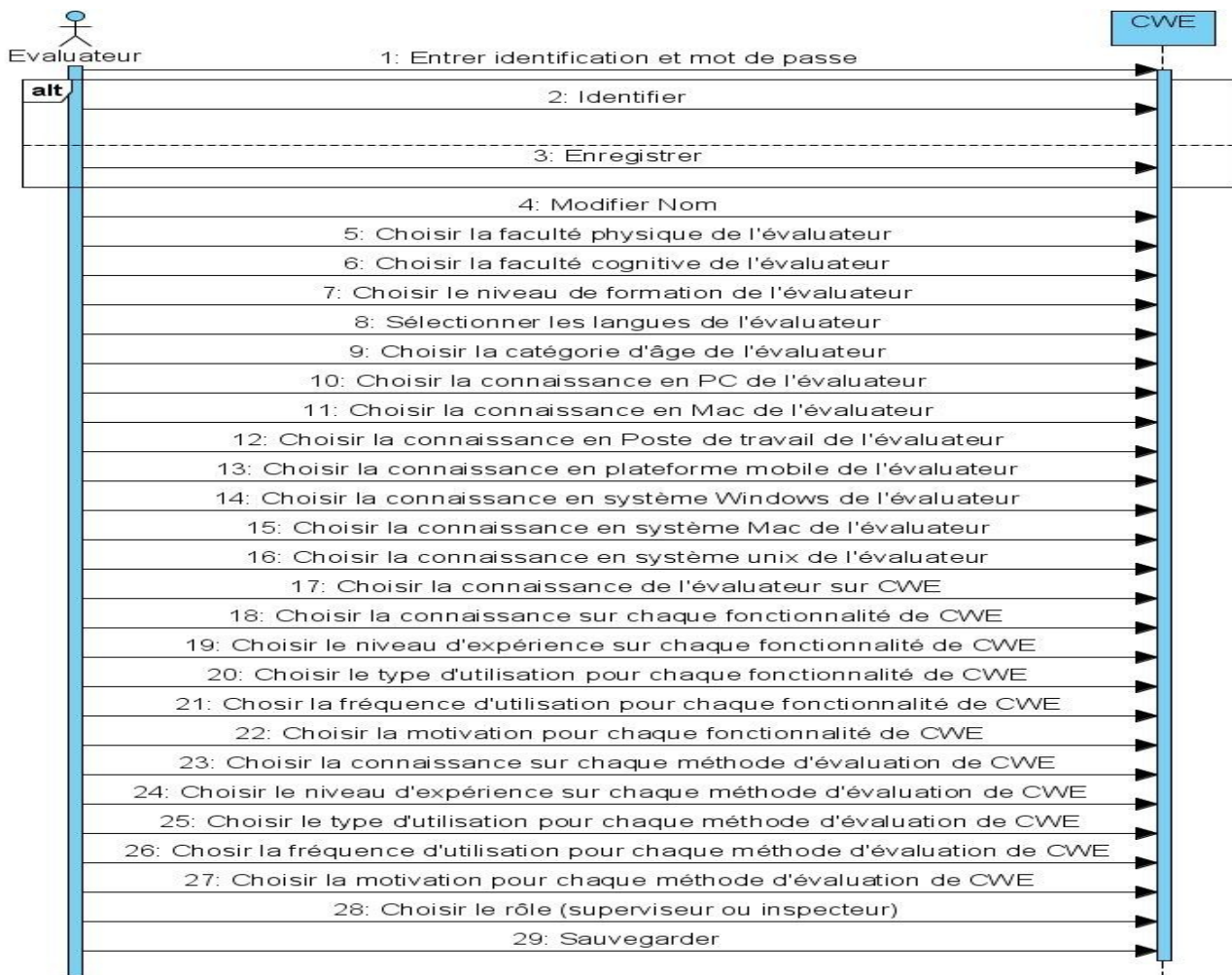


Figure 3.13 : Identification de l'évaluateur

En plus, chaque évaluateur doit fournir aussi sa connaissance et ses expériences sur les différentes méthodes supportées par CWE. Ainsi, il doit fournir sa connaissance, sa fréquence d'utilisation, son niveau d'expérience, son type d'utilisation et sa motivation pour chaque méthode.

Le déroulement de cette opération est décrit par la figure 3.13.

### 3.4.8 Inspection

L'inspection de tâches, décrite par la figure 3.14, permet aux inspecteurs d'évaluer l'interface et d'enregistrer selon le modèle du paragraphe §3.3.5 les problèmes d'utilisabilité qu'ils observent. Pour cela, ils doivent parcourir l'interface selon les suites d'actions définies par le superviseur durant l'analyse de tâche. A chaque action, ils évaluent l'interface en répondant aux suites d'actions correspondant à la méthode d'évaluation choisie. La réponse à une question doit être parmi les suivantes [Bligard & Osvalder, 2007] :

- Oui : la réponse est affirmative,
- Probablement oui : la réponse est probablement affirmative,
- Non : la réponse est négative,
- Probablement non : la réponse est probablement négative,
- Inconnue : il est impossible de décider entre Oui et Non.

A chaque réponse donnée, les inspecteurs doivent fournir une justification qui doit faire référence dans la mesure du possible à la cognition de l'utilisateur ; c'est-à-dire en référence à ses connaissances, ses expériences antérieures ou perceptions présumées du système.

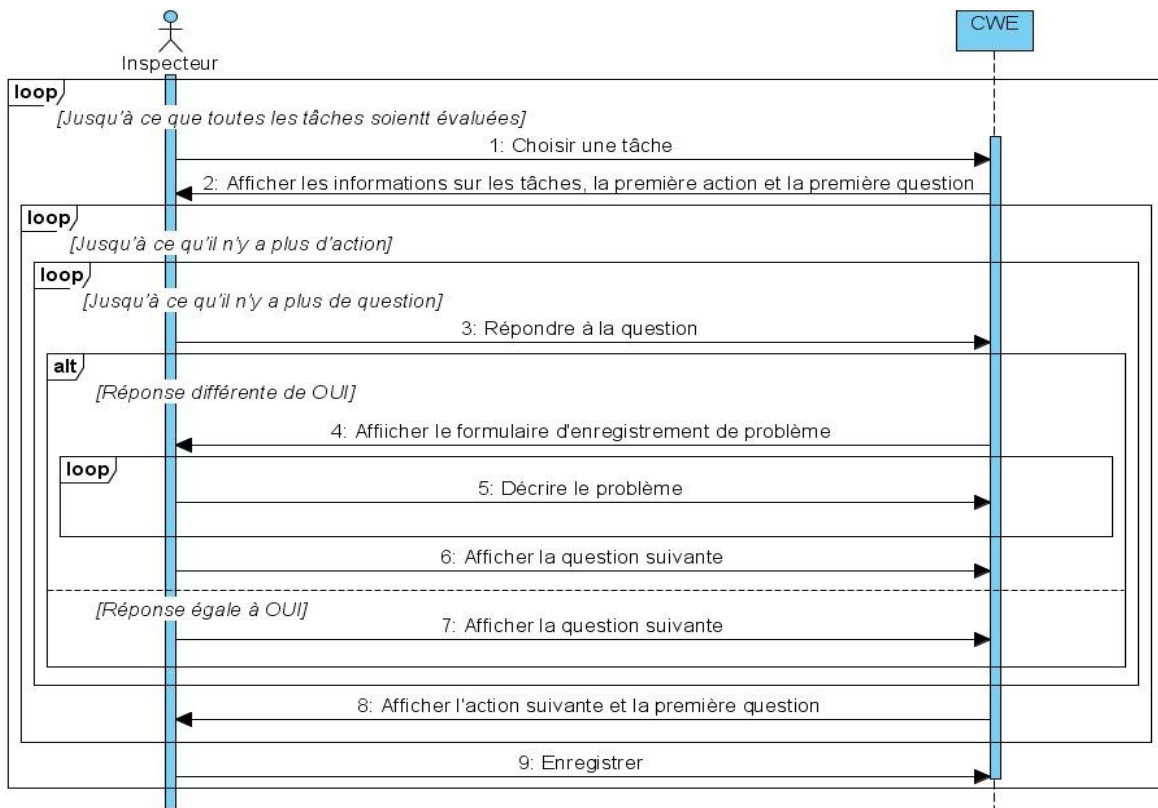


Figure 3.14 : Inspection de tâches

A la suite d'une réponse non affirmative (différente de Oui) qui signifie qu'on se trouve probablement en présence d'un problème d'utilisabilité, les inspecteurs doivent décrire ce problème en fournissant les informations suivantes : le numéro du problème, la référence du problème, la découverte du problème, la description du problème, le pourcentage des utilisateurs qui peuvent rencontrer le problème, la fréquence du problème, la gravité du problème, un commentaire ou solution.

### 3.4.9 Analyse des résultats

L'objectif de l'analyse des résultats est de produire un rapport de description des problèmes d'utilisabilité. Pour cela, le superviseur doit analyser les données fournies par les inspecteurs durant la phase d'inspection. L'analyse des résultats doit produire entre autre la synthèse des résultats de chaque inspecteur et la synthèse des problèmes d'utilisabilité. La première synthèse décrit les problèmes d'utilisabilité trouvés par chaque inspecteur, la façon dont il les a trouvés et la façon dont il les a jugés (voir tableau 3.4). La deuxième synthèse concerne les problèmes d'utilisabilité selon leur origine et leur catégorie (voir tableau 3.5).

En plus de ces deux tableaux de bord, il est intéressant de confronter les résultats fournis par les inspecteurs entre eux (voir tableau 3.6). L'objectif étant de classer les inspecteurs dont les résultats se ressemblent et d'isoler le ou les évaluateurs ayant des résultats excentrique s'il y en a. Dans l'état actuel des travaux, la façon la plus simple de procéder à cette confrontation est de calculer la distance par rapport à la moyenne de chaque réponse de l'évaluateur.

**Tableau 3.4 Synthèse des problèmes d'utilisabilité par inspecteur**

Problème	Question	Gravité	Justification	Pourcentage	Fréquence	Catégorie	Solution
Problème1							
Problème2							
...							

**Tableau 3.5 Synthèse des problèmes d'utilisabilité**

Catégorie	Question	Moyenne Gravité	Recommandations				
			Inspecteur <sub>1</sub>	Inspecteur <sub>2</sub>	...	Inspecteur <sub>n-1</sub>	Inspecteur <sub>n</sub>
					...		
					...		

A partir de ces différents tableaux de bord et des informations relatives au système, aux utilisateurs, aux tâches et aux inspecteurs, le superviseur peut effectuer les activités suivantes pour produire de bons rapports d'évaluation (selon les recommandations de Jeffries [1994]) : détecter les faux problèmes, détecter les erreurs des évaluateurs, évaluer les partis pris des évaluateurs, détecter les recommandations conflictuelles. Par exemple, si la distance totale des réponses d'un inspecteur s'écarte beaucoup de la moyenne, l'inspecteur doit vérifier s'il s'agit d'un partie pris de sa part, ou si c'est à cause de son expérience en évaluation. En revanche, si la distance totale des réponses d'un

inspecteur se rapproche de la moyenne et qu'il y a une réponse qui s'écarte de celle-ci, l'inspecteur doit vérifier s'il a commis des erreurs.

**Tableau 3.6 Confrontation des résultats des inspecteurs**

	Inspecteur <sub>1</sub>	Inspecteur <sub>2</sub>	...	Inspecteur <sub>n-1</sub>	Inspecteur <sub>n</sub>
Q <sub>1</sub>			...		
Q <sub>2</sub>			...		
Q <sub>m-1</sub>			...		
Q <sub>m</sub>			...		
Distance			...		

### 3.5 Conclusion

Nous avons présenté dans ce chapitre une spécification de environnement CWE (Cognitive Walkthrough Environment) qui se compose d'un ensemble d'outils d'aide pour l'évaluation des IHM basée sur la méthode Cognitive Walkthrough. Cette environnement couvre le processus d'évaluation : de la préparation à l'analyse des résultats. Dans la première partie du chapitre, nous avons présenté les deux rôles des évaluateurs durant le processus d'évaluation : superviseur et inspecteur. Nous avons présenté aussi les fonctionnalités fournies par l'environnement CWE.

Dans la deuxième partie, nous avons présenté une description conceptuelle des objets du monde de l'évaluation des IHM basée sur la méthode Cognitive Walkthrough. Nous avons présenté les concepts d'artefact et d'utilisateur : l'artefact est l'objet avec qui l'utilisateur interagit. On a vu aussi le concept de tâche comme un concept qui motive l'utilisateur à interagir avec l'artefact. Nous avons présenté aussi un modèle unique pour la plupart des méthodes Cognitive Walkthrough décrites dans le chapitre 2. Les représentations de plusieurs des onze méthodes selon ce modèle ont ainsi été décrites dans ce chapitre. Dans la troisième partie, nous avons présenté le comportement dynamique de CWE. Le déroulement d'une session d'évaluation basée sur notre modèle est ainsi décrit.

Dans le chapitre suivant, nous allons présenter un premier prototype de CWE.



# Chapitre 4 - Maquettage de CWE, environnement d'aide à l'évaluation de systèmes interactifs, basé sur la méthode Cognitive Walkthrough

---

## Plan du chapitre

4.1- Introduction.....	2
4.2- Objectifs architecturaux et contraintes.....	2
4.2.1 - Exigences logicielles.....	2
4.2.2 - Exigences spéciales de développement.....	2
4.3- Architecture de CWE.....	4
4.3.1 - Décisions architecturales.....	4
4.3.2 - Vue cas d'utilisation.....	6
4.3.3 - Vue logique.....	21
4.4- Conclusion.....	32

---

## 4.1 Introduction

Dans ce chapitre, nous proposons une maquette de CWE (Cognitive Walkthrough Environment) dont la spécification a été décrite dans le chapitre précédent. Une première version de cette maquette a été présentée dans [Mahatody et al., 2009]. Le maquettage consiste à façonner l'environnement CWE et à lui donner une forme et une architecture répondant globalement aux besoins et exigences formulés dans le chapitre précédent. Ce chapitre décrit donc principalement l'architecture et l'interface homme-machine de l'environnement CWE. Comme dans le chapitre précédent, nous utilisons la notation UML pour décrire cette maquette. Dans un premier temps, les objectifs architecturaux et les différentes contraintes de CWE sont expliquées. Ensuite, l'architecture de CWE est présentée. Enfin, avant de conclure ce chapitre, le mécanisme de persistance utilisé dans CWE est expliqué.

## 4.2 Objectifs architecturaux et contraintes

Cette section va décrire les exigences logicielles et les objectifs qui ont un impact significatif sur l'architecture de l'environnement CWE. Ce sont des besoins ou fonctionnalités qui ne peuvent pas être capturés par le diagramme des cas d'utilisations tels que les différentes contraintes matérielles, logicielles et de gestion. En outre, les exigences spéciales applicables à la conception et à la stratégie d'implémentation seront aussi décrites dans cette section.

### 4.2.1 Exigences logicielles

Les exigences logicielles concernent la sûreté, la sécurité, la confidentialité, la portabilité, et l'utilisation du produit fini. Pour l'environnement CWE, nous avons décidé qu'il soit multilingue ; ce qui nous oblige donc à externaliser les différents messages de CWE.

Il est à noter aussi que chaque utilisateur doit s'identifier par un identifiant et un mot de passe avant d'utiliser l'environnement ; il n'est pas possible de changer d'identité en cours de session.

### 4.2.2 Exigences spéciales de développement

Pour le développement de la maquette, Java est utilisé comme langage de programmation et MySQL comme système de gestion des bases des données. Java est à la fois un langage de programmation orienté objet et un environnement d'exécution informatique portable. Il permet donc d'écrire des logiciels portables sur plusieurs systèmes d'exploitation tels que Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications. En plus il peut être utilisé sous licence GPL (General Public License).

Quand à MySQL, c'est le serveur de base de données le plus utilisé dans le monde actuellement. Il peut être utilisé sous licence GPL comme Java. En plus il est rapide, robuste et facile à utiliser et à administrer.

Pour les différentes modélisations, UML est utilisé comme dans chapitre précédent. Et pour la représentation architecturale, le modèle « 4+1 vues » de Kruchten [1995] (voir figure 4.1) est adopté.

Pour faciliter la compréhension de la description de l'architecture de l'environnement, une brève description du modèle de Kruchten est fournie ci-après.

Il existe plusieurs manières de voir un système ; Morand [2001] trouve par exemple cinq approches pour définir le concept d'architecture logicielle. Kruchten [1995] propose une description architecturale composée de quatre vues et une cinquième vue fournissant des cas d'utilisation et reliant les quatre premières (« 4+1 vues »).

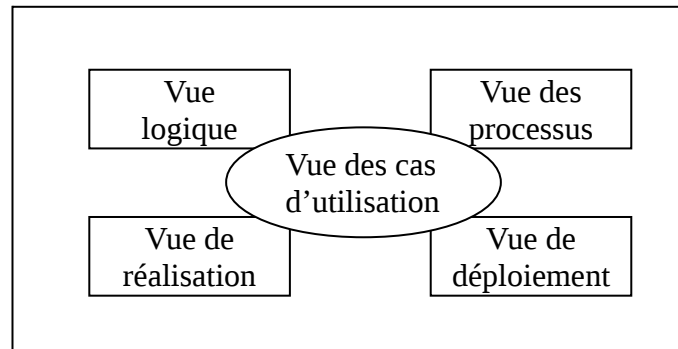


Figure 4.1 : Les "4+1" vues de Kruchten [1995]

La vue logique met l'accent sur la compréhension des fonctionnalités de l'application. C'est une vue de haut niveau qui modélise les éléments et mécanismes principaux de l'application. Elle permet donc d'identifier les différents éléments du domaine, ainsi que les relations et interactions entre ces éléments.

La vue des processus considère des aspects non fonctionnels tels que les performances, l'évolutivité et la capacité. Elle aborde les questions de concurrence, de distribution et de tolérance aux pannes. Elle représente le modèle de traitement du système. Elle capture les aspects concurrentiels, de synchronisation et de distribution de la vue logique.

La vue de réalisation décrit la décomposition du logiciel en couches et sous systèmes dans le modèle d'implémentation.

La vue de déploiement englobe les nœuds qui forment la topologie du système matériel sur lequel le système s'exécute. Elle se concentre sur la distribution, la communication et l'approvisionnement. Elle fournit toutes les configurations matérielles possibles en faisant la correspondance entre les composants de la vue de réalisation et ses configurations.

Tableau 4.1 : Les différentes diagrammes utilisables dans chaque vue [Muller & Gaertner, 2002]

	Vue des cas d'utilisation	Vue logique	Vue de réalisation	Vue de processus	Vue de déploiement
Diagramme de cas d'utilisation	X				
Diagramme de classes		X			
Diagramme d'objets	X	X			
Diagramme de séquence	X	X		X	
Diagramme de collaboration	X	X		X	
Diagramme d'états-transition	X	X		X	
Diagramme d'activité	X	X		X	
Diagramme de composants			X		X
Diagramme de déploiement					X
Diagramme de package		X			

En plus des quatre vues ci-dessus, une vue centrale est destinée à capturer les scénarios. Cette vue englobe les cas d'utilisation qui décrivent le comportement du système du point de vue des utilisateurs finaux et des autres intervenants.

En utilisant la notation UML, le tableau 4.1 montre les diagrammes UML qu'on peut utiliser pour chaque vue architecturale.

## 4.3 Architecture de CWE

Cette section décrit l'architecture de l'environnement CW. Dans un premier temps, les décisions architecturales prises pour satisfaire les objectifs architecturaux et les différentes contraintes décrits dans la section 2 sont listées. Puis la vue logique et la vue des cas d'utilisation sont décrites.

### 4.3.1 Décisions architecturales

Nous présentons ici les décisions principales prises pour la réalisation de CWE. Ces décisions concernent entre autre le multilinguisme, la persistance des objets et les IHM.

Pour le multilinguisme, nous utilisons la classe « ResourceBundle » de Java. Cette classe permet d'obtenir le message adéquat selon la langue utilisée. Nous implantons deux langues : le français et l'anglais. La classe « ResourceBundle » utilise des fichiers propriétés (.properties). Nous avons donc deux fichiers propriétés : cweMYRS (figure 4.2) pour la langue française et cweMYRS\_en (figure 4.3) pour la langue anglaise. Des extraits pour chaque fichier sont visibles dans ces deux figures.

```
ScenarioScreen.Scenario.nouveau=Nouveau scenario
ScenarioScreen.Ressource.nom=Nom
ScenarioScreen.Ressource.nouvelle=Nouvelle ressource
ScenarioScreen.Attribut.nouvel=Nouvel attribut
TaskScreen.Tache.nouvelle=nouvelle tache
UserScreen.Age.enfant=Enfant
UserScreen.Age.senior=personne agée
UserScreen.Utilisateur.nouvel=Nouvel utilisateur
```

Figure 4.2 : cweMYRS.properties (extrait)

```
ScenarioScreen.Scenario.nouveau=New scenario
ScenarioScreen.Ressource.nom=Name
ScenarioScreen.Ressource.nouvelle=New resource
ScenarioScreen.Attribut.nouvel=New attribute
TaskScreen.Tache.nouvelle=new task
UserScreen.Age.enfant=kid
UserScreen.Age.senior=Elderly
UserScreen.Utilisateur.nouvel=New user
```

Figure 4.3 : cweMYRS\_en.properties (extrait)

Rappelons que les objets persistants sont des objets qui peuvent survivre tout au long du processus qui les a créés, et qu'un objet persistant existe jusqu'à ce qu'il soit explicitement supprimé [Larman, 2005]. Nous avons besoin d'un service de persistance qui fournit des fonctionnalités pour supporter les objets persistants. Dans notre cas, exploitant une base de données relationnelles, ce service est aussi appelé un service de correspondance Objet-Relation. Ce service doit donc traduire

les objets enregistrement et les enregistrer dans une base de données, et traduire les enregistrements en objets lors de la récupération à partir d'une base de données.

Le fonctionnement de ce service est illustré en figure 4.4. La correspondance des types d'attribut d'une classe et des types de colonne d'une entité est donnée par le tableau 4.2. Tandis que la correspondance entre association de classes et association d'entité relation est décrite par la figure 4.5.

Tableau 4.2 : Correspondance des types d'attribut de classe et de types de colonne d'une entité.

Diagramme de classe	Diagramme Entité-Relation	Diagramme de classe	Diagramme Entité-Relation
Boolean	Bit (1)	Double	Double(10)
Byte	Tinyint (3)	Decimal	Integer
Byte[]	Binary(1000)	BigDecimal	Decimal(19)
Blob	Blob	Float	Float(10)
Char	Char(1)	Long	Bigint(19)
Character	Char(1)	Short	Smallint(5)
String	varchar(255)	Date	Date
Int	Integer(10)	Time	Time(7)
Integer	Integer(10)	Timestamp	Timestamp(7)

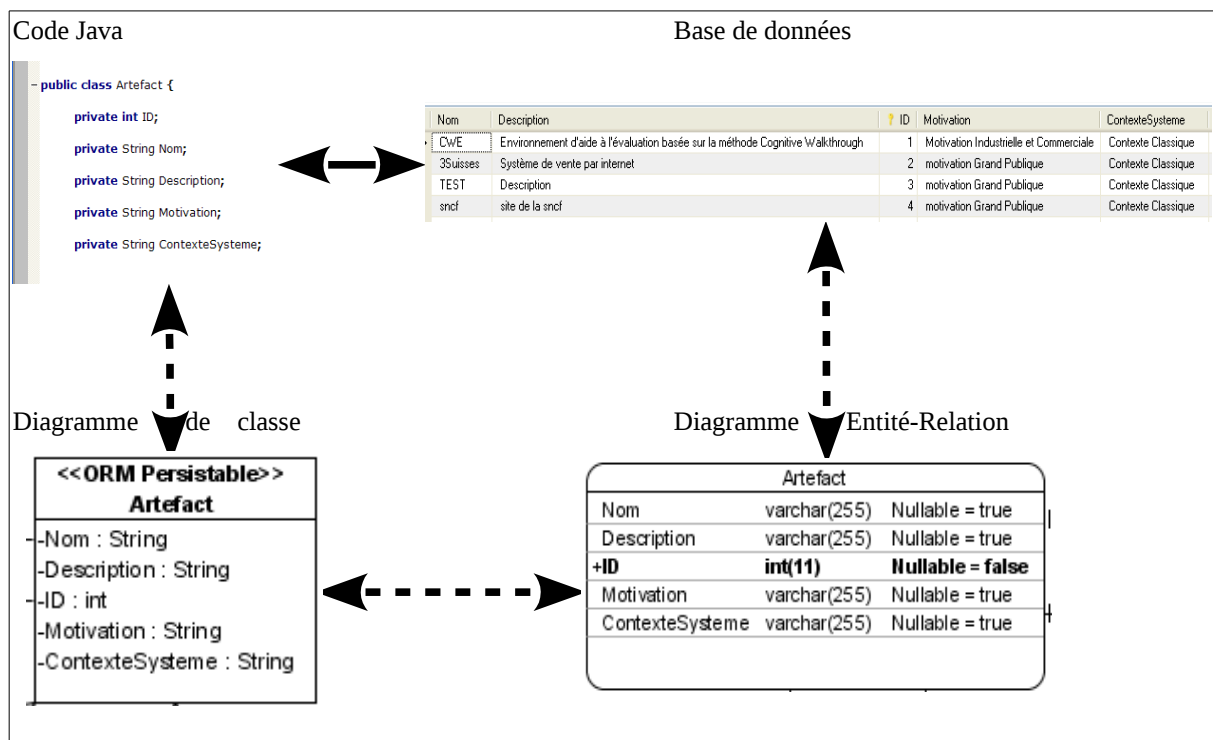


Figure 4.4 : Fonctionnement du service de persistance

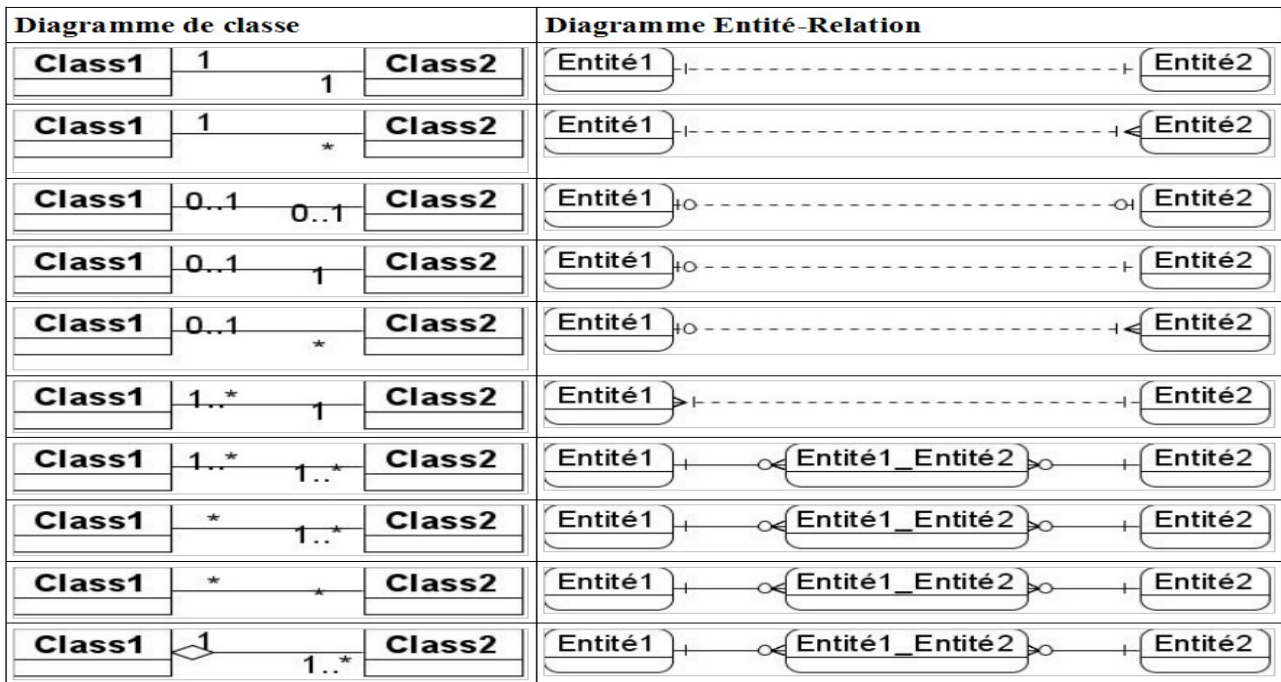


Figure 4.5 : Correspondance entre association de classes et association d'entité relation.

### 4.3.2 Vue cas d'utilisation

Dans cette section, nous allons présenter quelques cas d'utilisation significatifs qui illustrent le déroulement de l'évaluation en partant de la phase de préparation jusqu'à la phase d'analyse de résultat. Ces cas d'utilisation significatifs ainsi que les relations entre eux sont décrits dans le diagramme de la figure 4.6. Ils sont décrits dans le chapitre 3. Nous nous contenterons de présenter les interfaces homme-machines correspondantes.

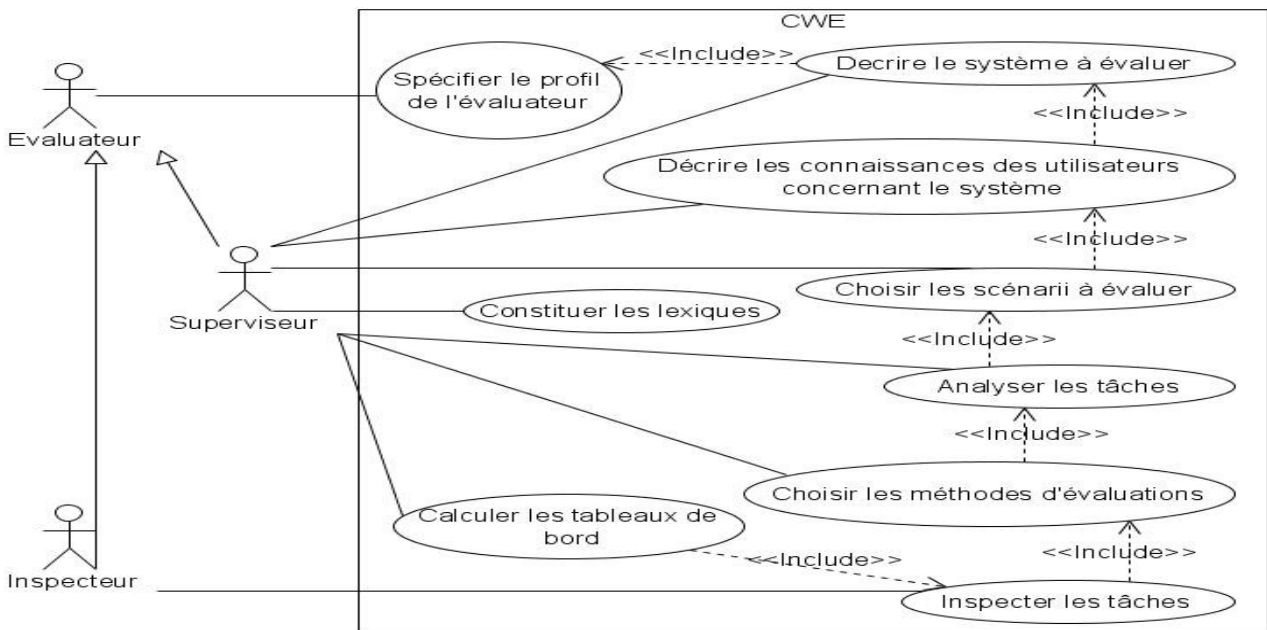


Figure 4.6 : Cas d'utilisation significatifs

### 4.3.2.1 Cas d'utilisation : « Décrire le système à évaluer »

Ce cas d'utilisation utilise l'onglet « Système » de l'écran « Ecran de description de système » (figure 4.7). Cet écran est obtenu à partir du menu « Préparation », item « Spécification ». L'onglet « Système » propose un formulaire permettant de saisir les différentes informations sur l'application ou le système à évaluer. Cet onglet possède un formulaire pour l'identité de l'application (nom et description). Il contient aussi une liste de domaines connus par l'environnement CWE offrant au superviseur de sélectionner un domaine dans la liste. En outre à droite de la liste, on trouve un formulaire permettant de saisir un nouveau domaine si le domaine de l'application à évaluer ne se trouve pas dans la liste.

L'onglet « Système » permet aussi de choisir la motivation et le contexte d'utilisation du système. Il permet entre autre d'ajouter de rôles et fonctionnalités au système à évaluer.

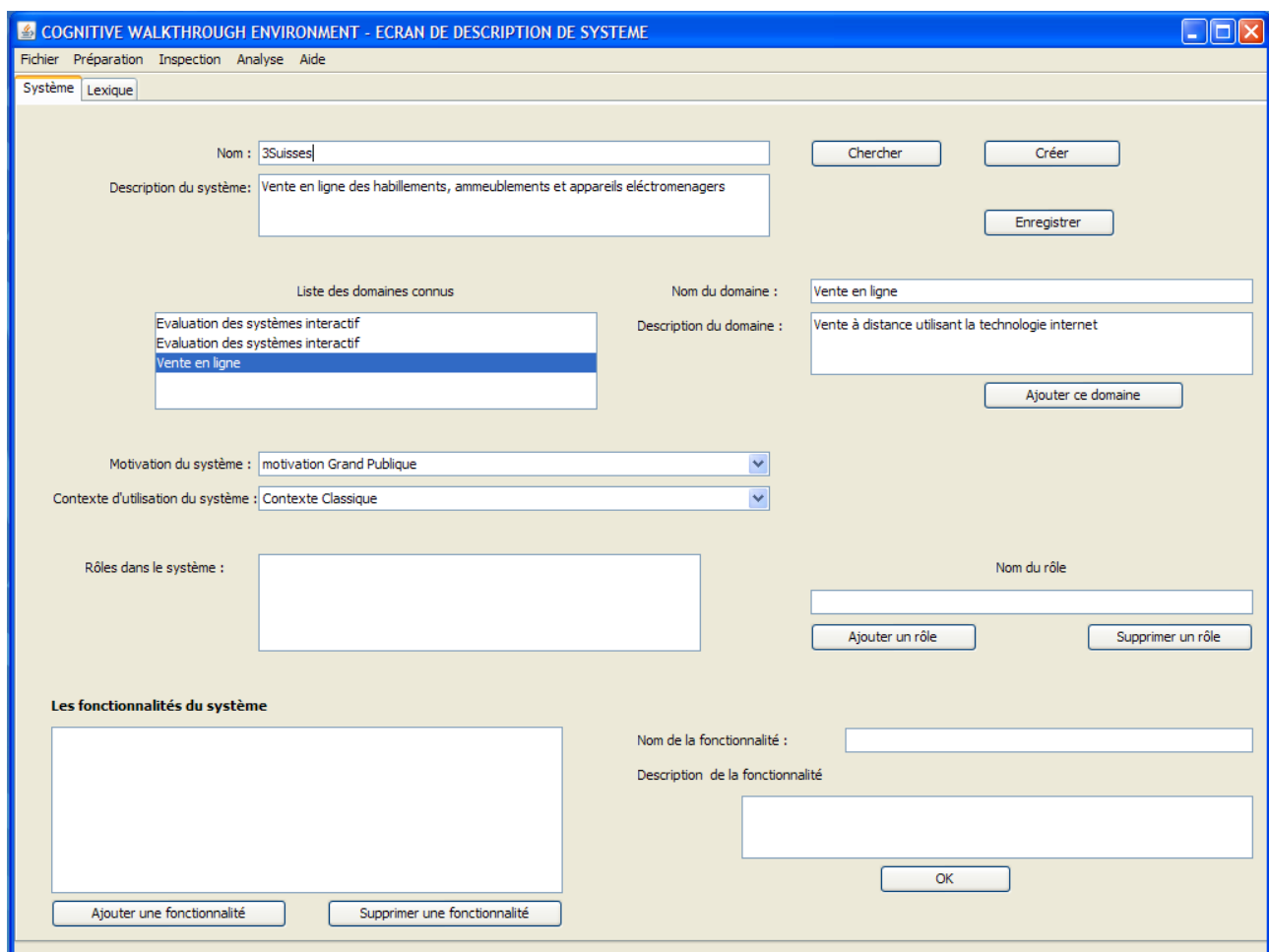


Figure 4.7 : Onglet de description du système à évaluer

### 4.3.2.2 Cas d'utilisation : « Constituer les lexiques »

Ce cas d'utilisation utilise l'onglet « Lexique » de l'écran d'identification du système (voir figure 4.8). Cet onglet permet de choisir le domaine sur lequel on constitue les lexiques. Ainsi, il permet d'ajouter de lexiques dans la liste des lexiques. A droite de cette liste, on trouve un formulaire permettant de définir le lexique et d'ajouter de notions et de comportement.

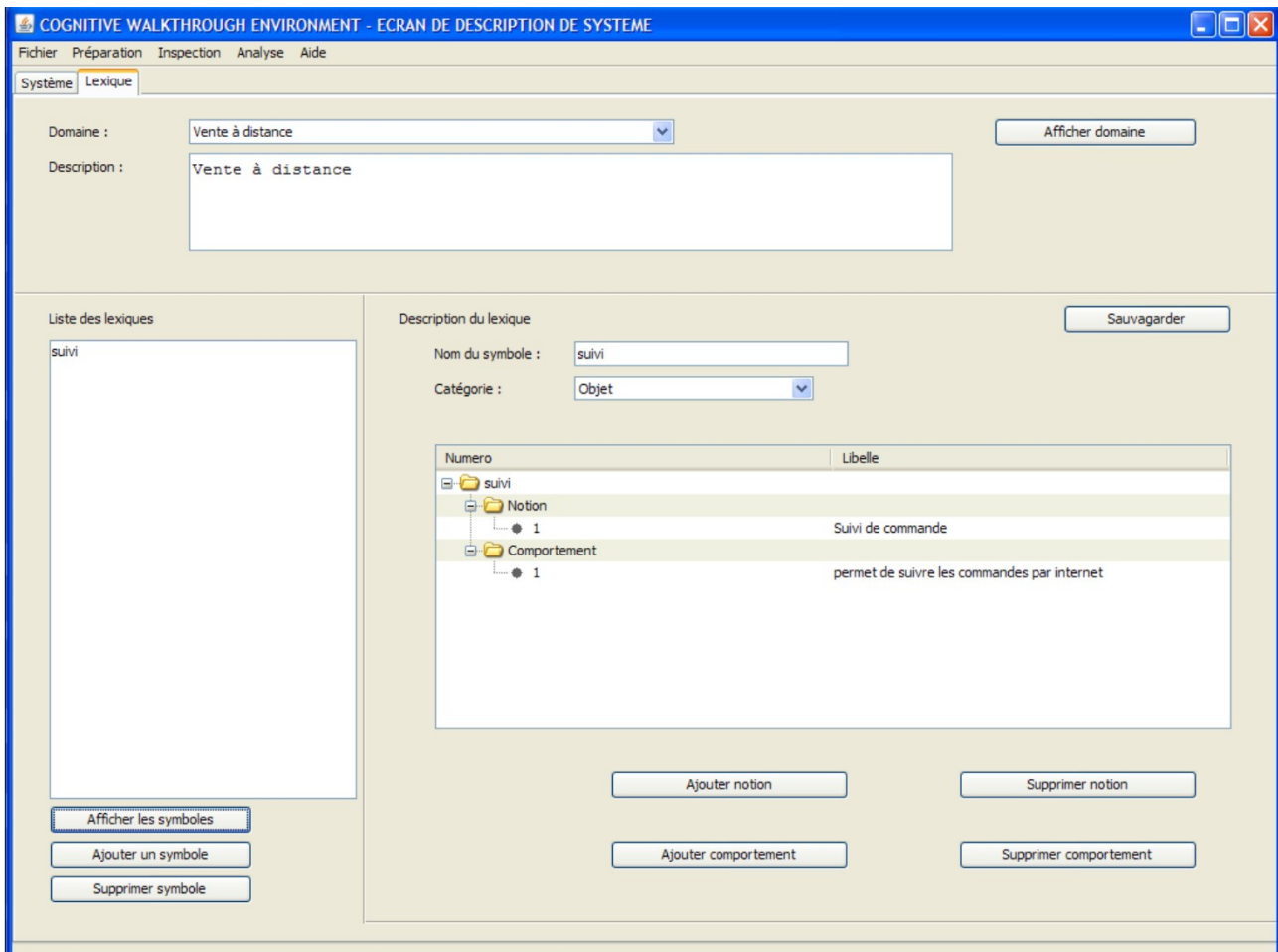


Figure 4.8 : Onglet de constitution des lexiques

### 4.3.2.3 Cas d'utilisation : « Identifier les utilisateurs »

Ce cas d'utilisation utilise l'onglet « Identité » de l'écran « Ecran d'identification utilisateur » (figure 4.9). Cet écran possède trois onglets : l'onglet « Identité », l'onglet « Connaissance informatique » et l'onglet « Connaissance de système ». L'onglet « Identité » permet de chercher le système à évaluer et d'y ajouter un utilisateur. A gauche on trouve la liste des utilisateurs et à droite on trouve un formulaire permettant de saisir les informations concernant l'utilisateur sélectionné dans la liste : niveau d'étude, langue, etc.



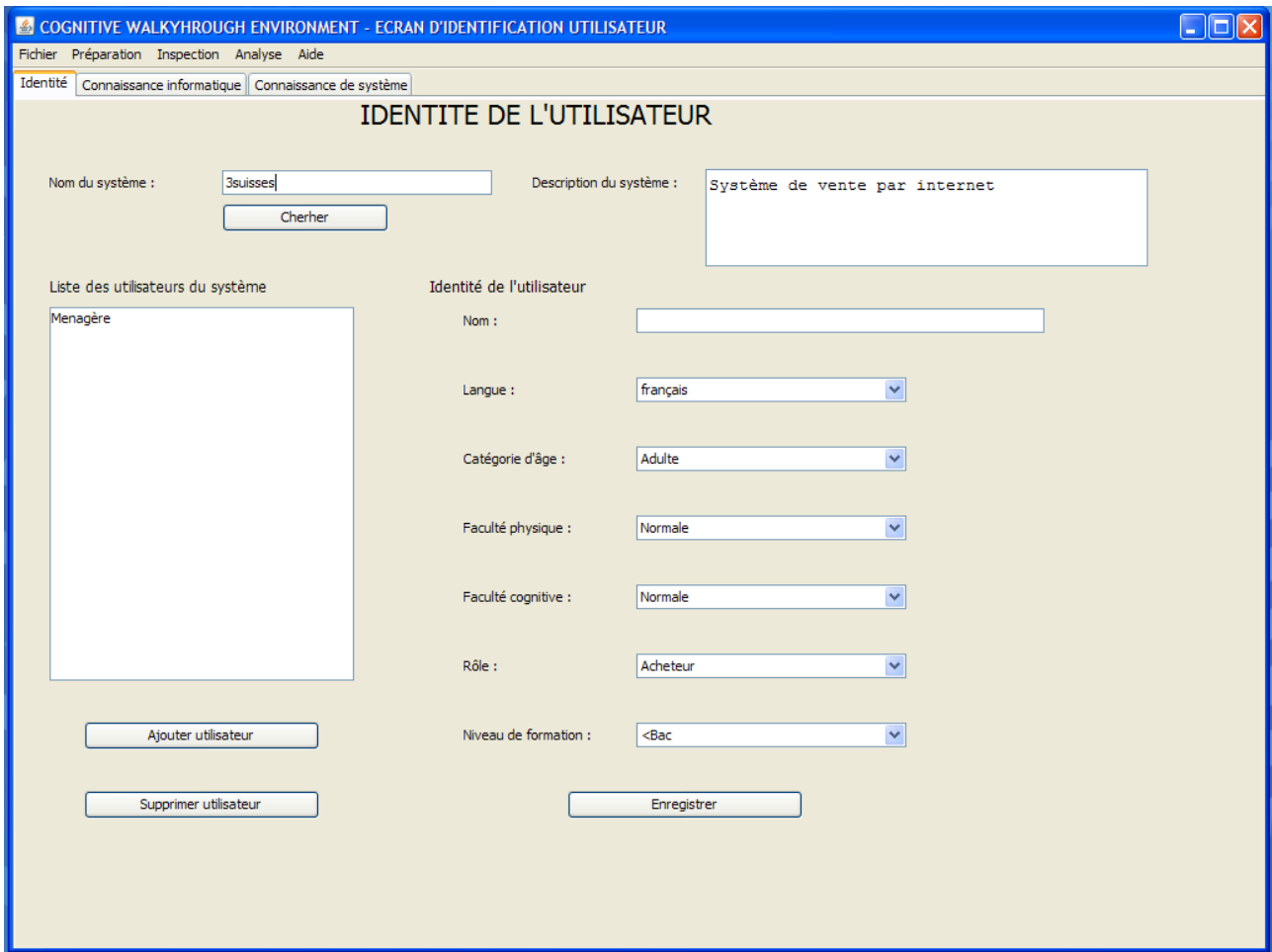


Figure 4.9 : Onglet d'identification de l'utilisateur

#### 4.3.2.4 Définir les connaissances informatique des utilisateurs

Ce cas d'utilisation utilise l'onglet « connaissance informatique » de l'écran d'identification de l'utilisateur (voir figure 4.10). Sur cet onglet, on trouve à gauche la liste des utilisateurs où on peut sélectionner un utilisateur et à droite un formulaire permettant de spécifier les connaissances informatique de l'utilisateur sélectionné dans la liste.

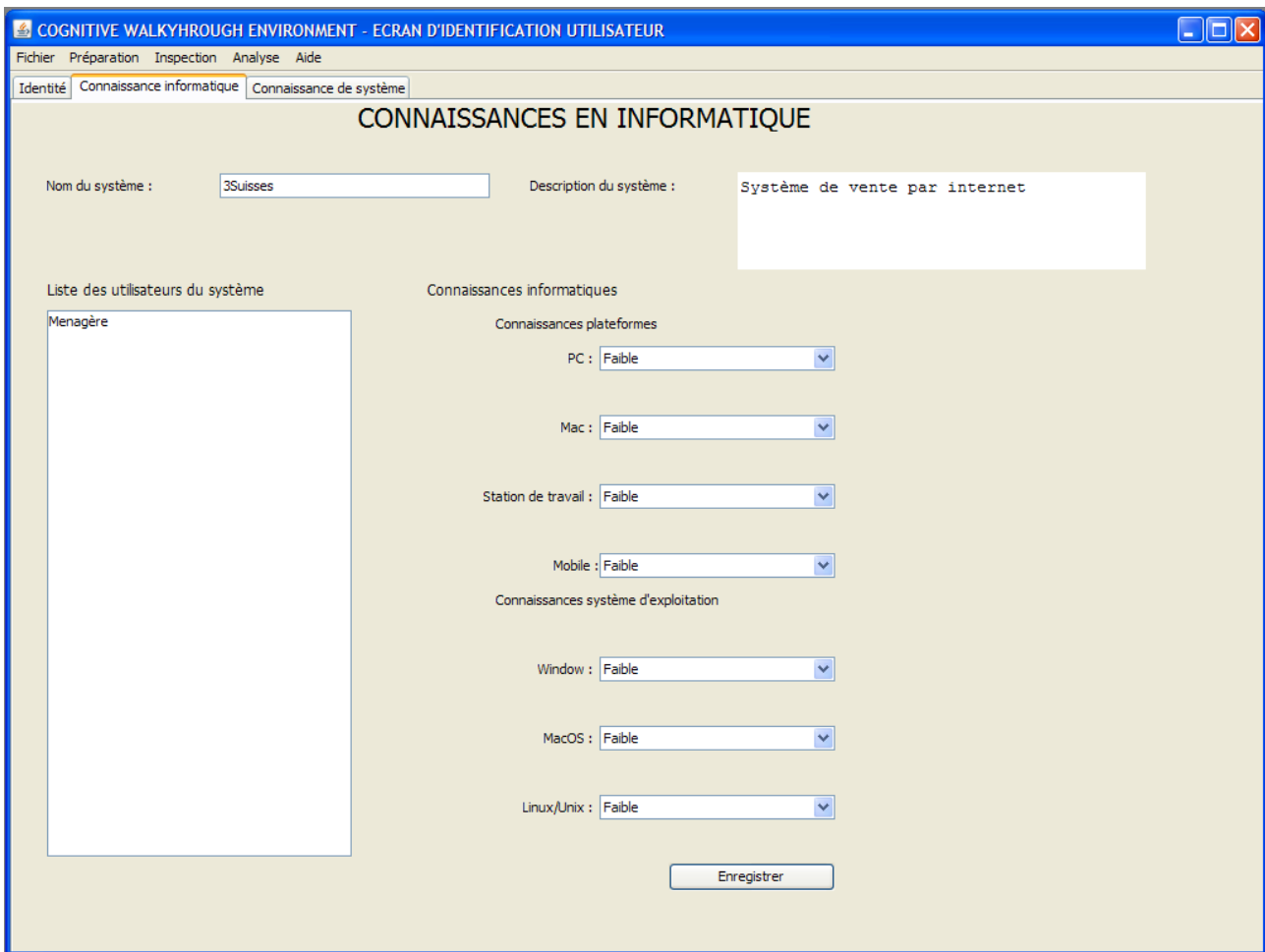


Figure 4.10 : Onglet de description des connaissances en informatique de l'utilisateur

### 4.3.2.5 Cas d'utilisation : « Décrire les connaissances des utilisateurs concernant le système »

Ce cas d'utilisation utilise l'onglet « Connaissance du système » de l'écran d'identification de l'utilisateur (voir figure 4.11). Sur cet onglet, on trouve à gauche la liste des utilisateurs où on peut sélectionner un utilisateur. Puis à droite, on peut spécifier la connaissance de l'utilisateur sélectionné concernant le domaine. On y trouve aussi un tableau à six colonnes permettant de définir les connaissances et les expériences de l'utilisateur sélectionné dans la liste de des utilisateurs. La première colonne devrait contenir les différentes fonctionnalités du système. Sur les autres colonnes le superviseur doit spécifier le niveau d'expertise, la fréquence d'utilisation, le type d'utilisation, la motivation et la connaissance de l'utilisateur sélectionné concernant la fonctionnalité de chaque ligne du tableau.

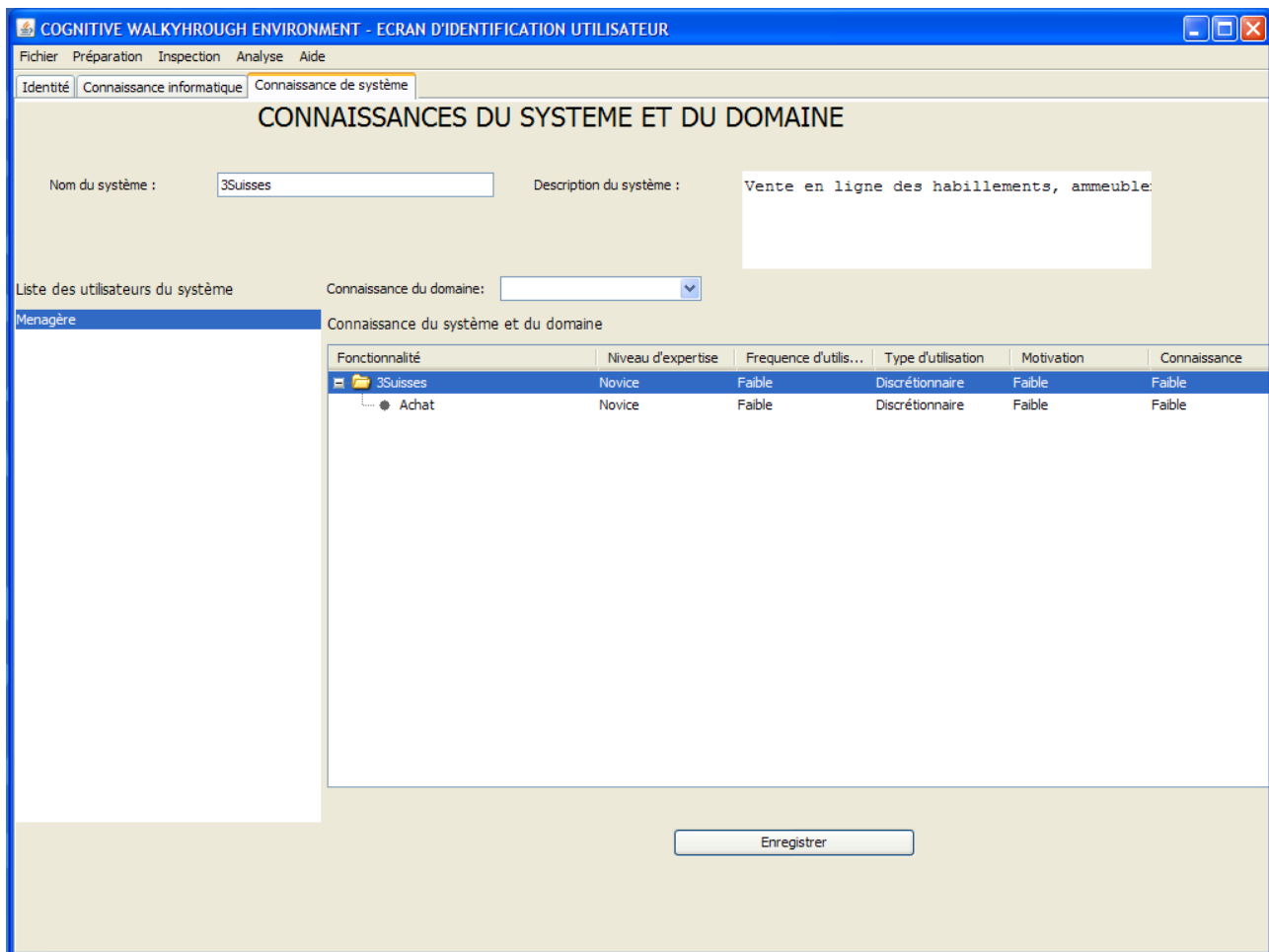


Figure 4.11 : Onglet de description des connaissances des utilisateurs concernant le système

#### 4.3.2.6 Cas d'utilisation : « Spécifier le profil de l'évaluateur »

Ce cas d'utilisation utilise l'onglet « Identification de l'écran « Ecran d'identification évaluateur » (figure 4.12). Cet écran est obtenu à partir du menu « Fichier », item « Profil évaluateur ». Il possède quatre onglets : « Identification », « Connaissance en informatique », « Expérience en CW » et « Expérience en CWE ». L'onglet « Identification » propose un formulaire permettant de saisir les nom et prénom de l'évaluateur connecté à l'environnement CWE, son rôle : inspecteur ou évaluateur, son nom d'identification et son mot de passe.

L'onglet « Connaissance en informatique » (voir figure 4.13) affiche le nom de l'évaluateur connecté à l'environnement CWE. On peut y spécifier les connaissances en informatiques de cet évaluateur.

L'onglet « Expérience en CW » (voir figure 4.14) affiche aussi le nom de l'évaluateur connecté à l'environnement , il permet de saisir les connaissances de l'évaluateur concernant les différentes variantes de la méthode Cognitive Walkthrough paramétrées dans CWE. Ainsi, on y trouve un tableau à six colonnes. La première colonne montre les méthodes déjà paramétrées dans l'environnement CWE. Les cinq autres colonnes permettent à l'évaluateur connecté de spécifier son niveau d'expertise, sa fréquence d'utilisation, son type d'utilisation, sa motivation et sa connaissance concernant la méthode de chaque ligne du tableau.

COGNITIVE WALKTHROUGH ENVIRONMENT - ECRAN D'IDENTIFICATION EVALUATEUR

Fichier Préparation Inspection Analyse Aide

Identification Connaissance en informatique Experience en CW Experience en CWE

### INFORMATION D'IDENTIFICATION

Nom et prénom : MAHATODY Thomas

Rôle : Inspecteur

Identification : thomas

Mot de passe :

Mot de passe (retaper):

Afficher Enregistrer

Figure 4.12 : Onglet d'identification de l'évaluateur

COGNITIVE WALKTHROUGH ENVIRONMENT - ECRAN D'IDENTIFICATION EVALUATEUR

Fichier Préparation Inspection Analyse Aide

Identification Connaissance en informatique Experience en CW Experience en CWE

### EXPERIENCE EN INFORMATIQUE

Nom et prénoms : MAHATODY Thomas

Connaissance sur les plateformes

PC : Bonne

Mac : Bonne

Poste de travail : Bonne

Mobile : Bonne

Système d'exploitation

Window : Bonne

MacOS : Bonne

Linux/Unix : Bonne

Afficher Enregistrer

Figure 4.13 : Onglet de description de l'expérience en informatique de l'évaluateur

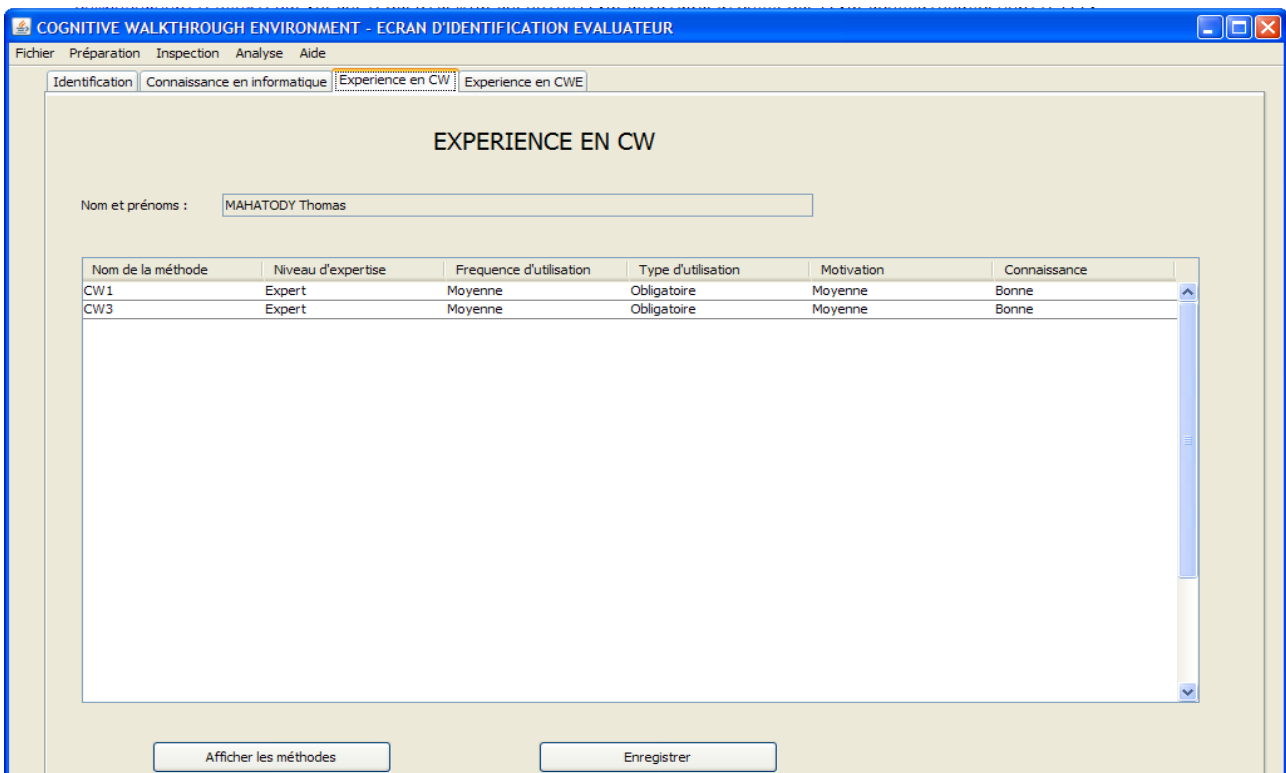


Figure 4.14 : Onglet de description des expériences en Cognitive Walkthrough

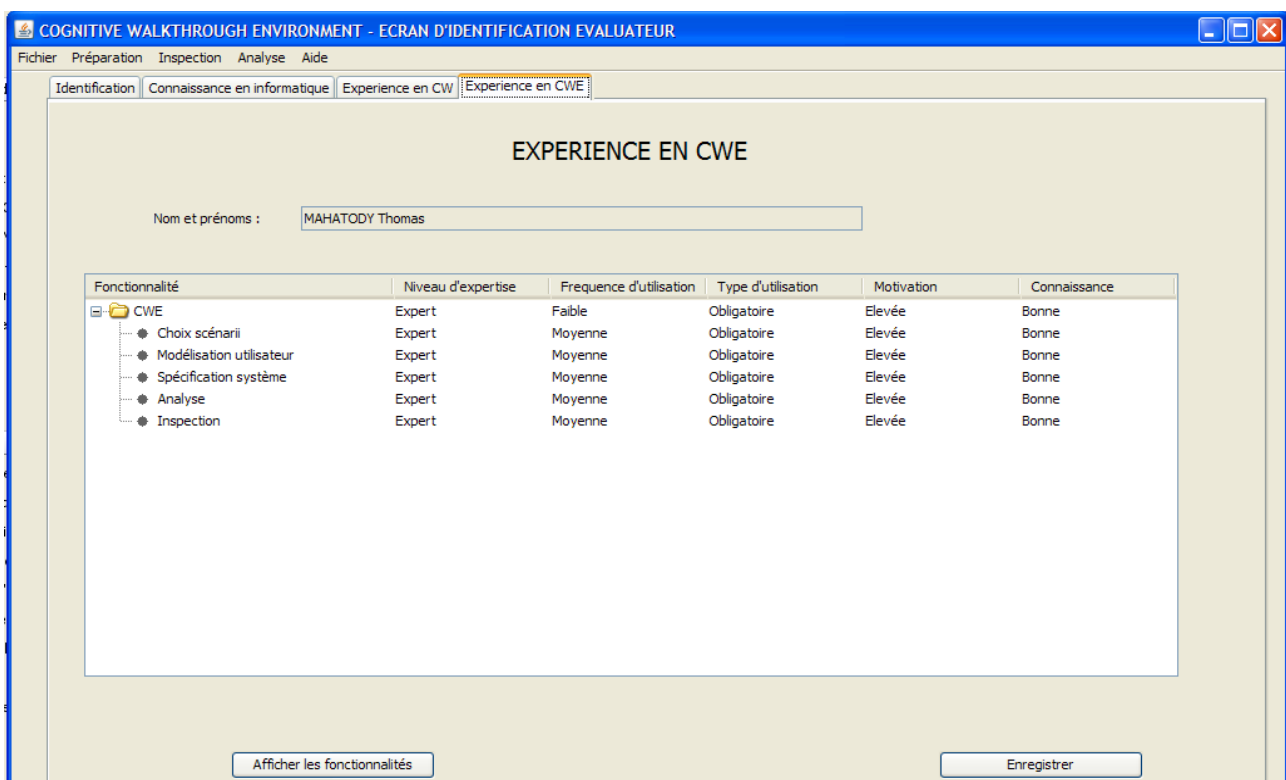


Figure 4.15 : Onglet de description des expériences en CWE

L'onglet « Expérience en CWE » (voir figure 4.15) permet de spécifier les expériences de l'évaluateur connecté à l'environnement CWE de spécifier ses expériences concernant les cinq

fonctionnalités principales de CWE. On y trouve un tableau de six colonnes; la première colonne affiche les cinq fonctionnalités de l'environnement. Les cinq autres colonnes permet à l'évaluateur de spécifier son niveau d'expertise, sa fréquence d'utilisation, son type d'utilisation, sa motivation et sa connaissance concernant la fonctionnalité de chaque ligne du tableau.

### 4.3.2.7 Cas d'utilisation : « Choisir les scénarii à évaluer »

Ce cas d'utilisation utilise l'écran « Ecran de description de scénario » visible en figures 4.16 et 4.17. Cet écran est obtenu à partir du menu « Préparation », item « Choix de scenarii ». Il possède deux onglets : l'onglet « Description scénario » et l'onglet « Description des ressources ». L'onglet « Description scénario » (voir figure 4.16) permet à l'utilisateur d'entrer le nom de l'application sur laquelle le scénario doit être exécuté. Cet écran contient un tableau intitulé « Liste des scenarii ». Ce tableau contient à gauche une colonne pour le nom du scénario et à droite autant de colonnes que le l'application possède de fonctionnalité. Pour créer un scénario, il faut cliquer sur le bouton « Ajouter un scénario ». Puis, un nouveau scénario portant le nom « Nouveau scénario » est inscrit dans le tableau « Liste des scenarii ». Ensuite, il faut cliquer sur ce nouveau scénario pour pouvoir le modifier sur le formulaire en bas du tableau. Et enfin, après avoir le modifier, on doit cliquer sur le bouton « Valider ». Après la validation, il faut cocher les fonctionnalités concernées par ce scénario dans les colonnes de droite du tableau « Liste des scenarii ».

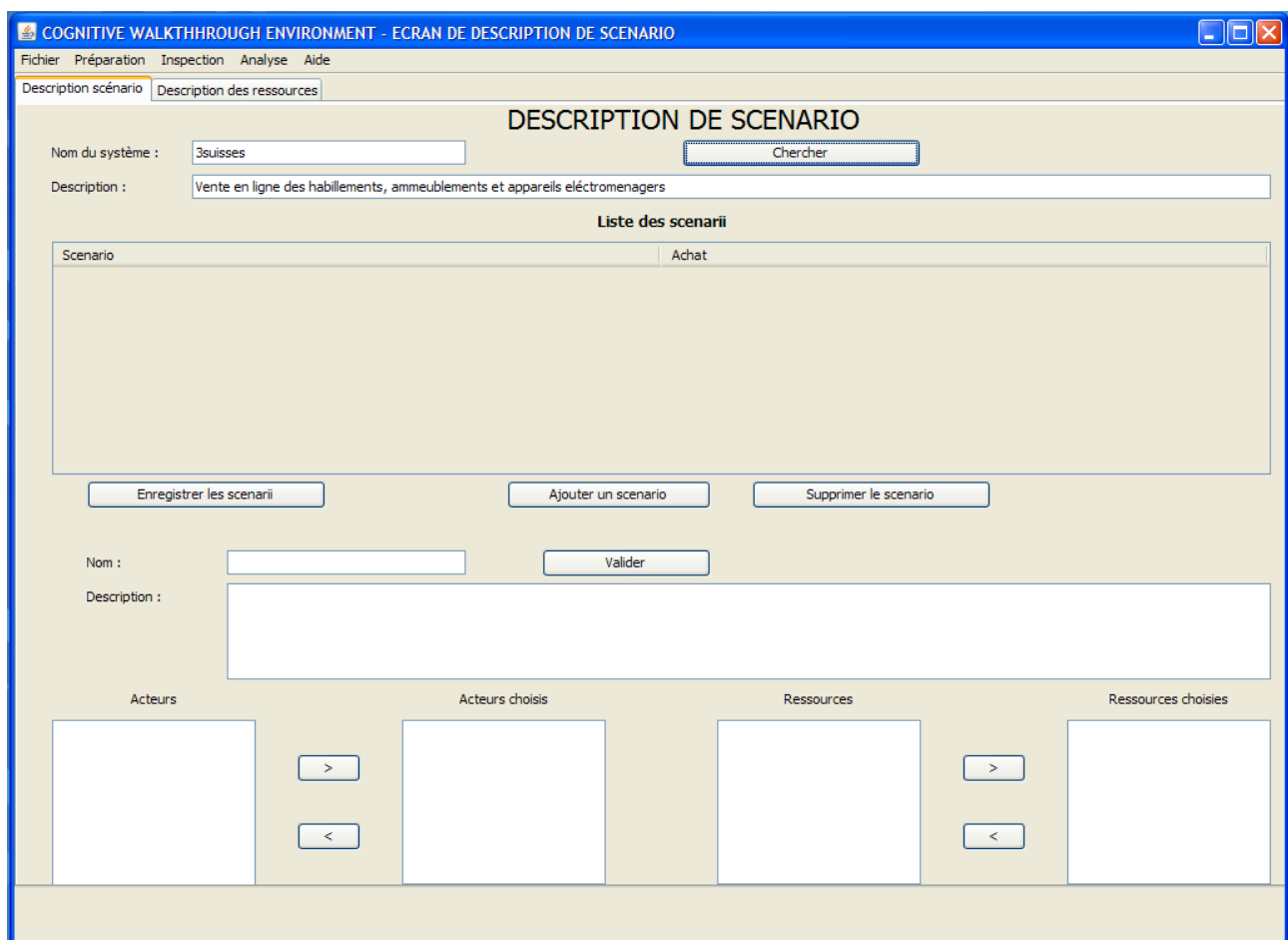


Figure 4.16 : Onglet de description des scenarii

L'onglet « Description des ressources » (voir figure 4.17) permet de définir les ressources que peuvent être requises lors de l'exploitation du système à évaluer. Cet onglet affiche donc le système en question et sa description. On y trouve un tableau à deux colonnes : la première colonne affiche les ressources et leurs attributs et la deuxième affiche la valeur de chaque attribut. Le superviseur peut ajouter une ressource et un attribut.

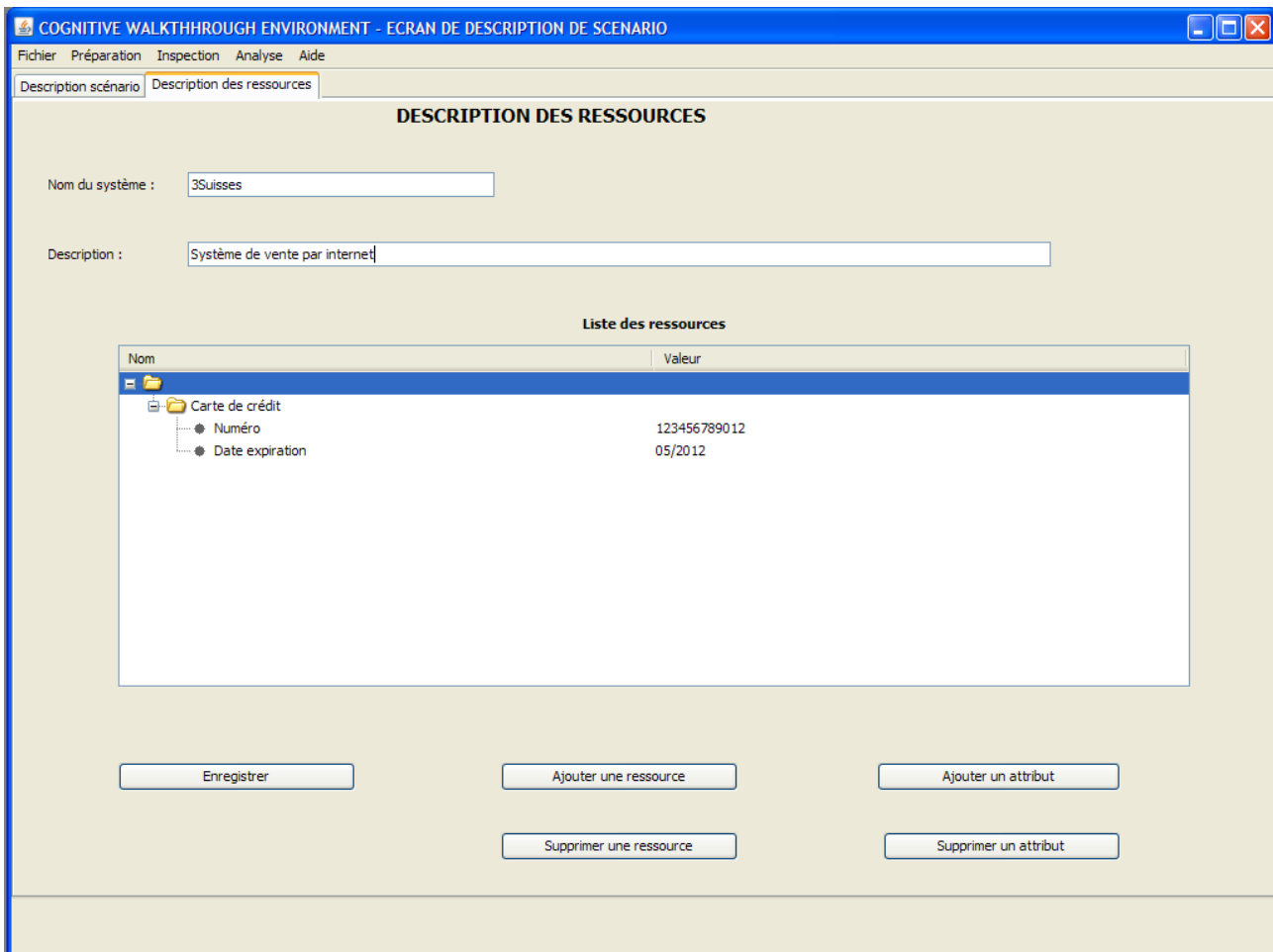


Figure 4.17 : Onglet de description des ressources

#### 4.3.2.8 Cas d'utilisation : « Analyser les tâches »

Ce cas d'utilisation utilise l'écran « Ecran d'analyse de tâche » visible en figure 4.18. Cet écran est obtenu à partir du menu « Préparation », item « Analyse de tâches ». Il comporte dans la partie haute, le nom du système et les descriptions des scénarii. Puis dans la partie basse, on trouve à gauche la liste hiérarchique des tâches et des actions et à droite le formulaire décrivant les caractéristiques des tâches.

Pour réaliser la décomposition d'un scénario en tâches et actions, il faut donner le nom du système concerné, et choisir un scénario. Le scénario sélectionné apparaît dans la liste des tâches. En le sélectionnant, on peut le décomposer en plusieurs tâches en cliquant autant de fois que le nombre de tâches nécessaires sur « Ajouter une tâche fille ». Puis chaque tâche peut être décomposée en plusieurs sous-tâches en utilisant aussi « Ajouter une tâche fille », ou en plusieurs sous-tâches individuelles ou sous-tâches collaboratives en utilisant respectivement « Ajouter une

tâche individuelle fille » ou « Ajouter une tâche collaborative fille ». Et enfin, chaque tâche individuelle ou tâche collaborative doit être décomposée en une suite d'actions en utilisant « Ajouter une action fille ».

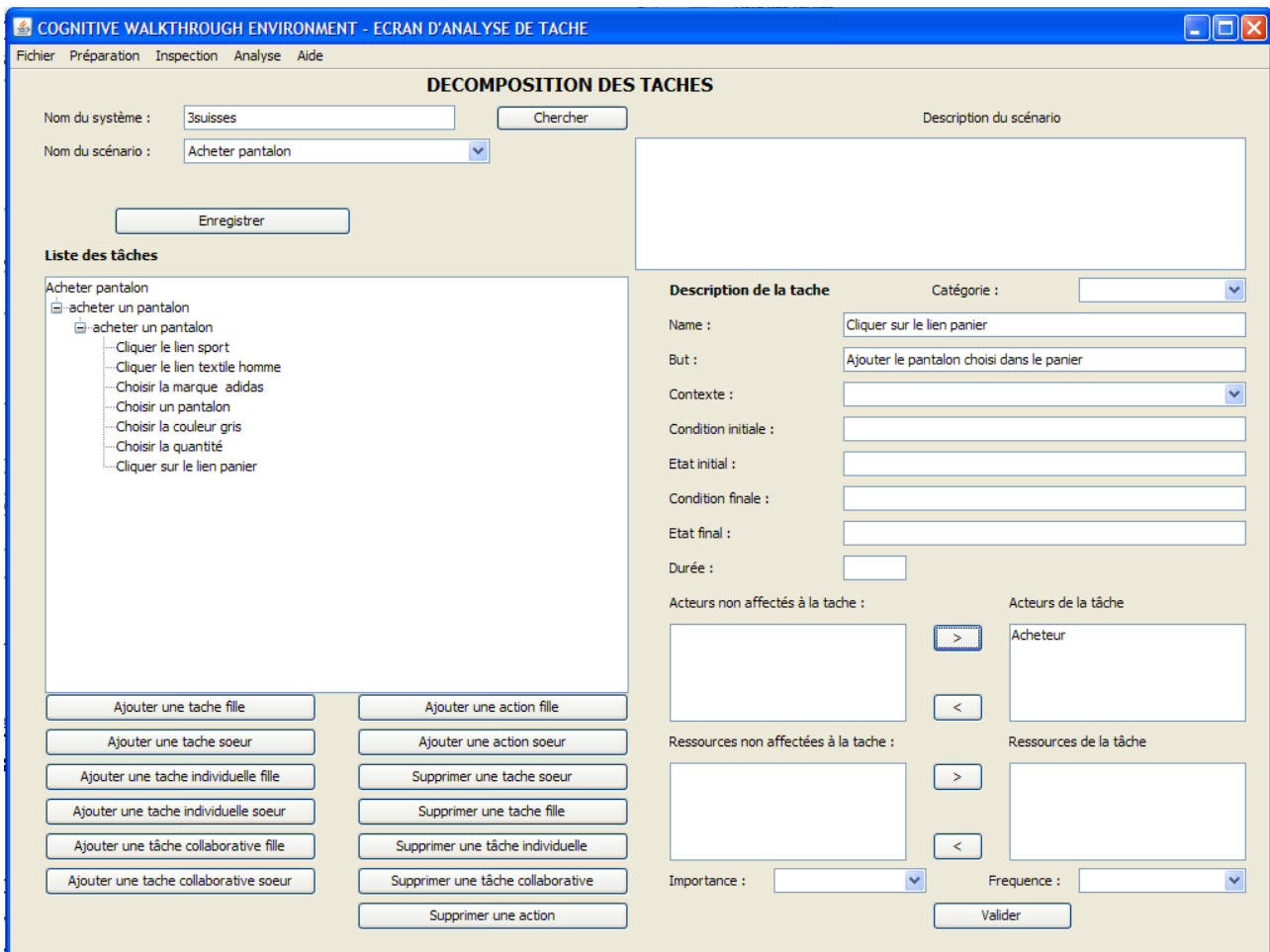


Figure 4.18 : Ecran d'analyse de tâche

### 4.3.2.9 Cas d'utilisation : « Choisir les méthodes d'évaluation »

Ce cas d'utilisation utilise l'écran « Choix de méthode » (figure 4.19). Cet écran est obtenu à partir du menu « Préparation » item « Choix de méthode ». Cet écran permet de chercher le système à évaluer à partir de son nom. La liste des scenarii relatifs au système trouvé apparaît dans une liste déroulante en bas du nom du système. La description du scénario choisi se trouve en bas de la liste déroulante des scenarii. On trouve aussi sur cet écran un tableau à six colonnes. Dans la première colonne de ce tableau on trouve les numéros des tâches à évaluer, relatives au scénario choisi. Sur la deuxième on trouve le nom de la tâche. La troisième permet au superviseur de désigner l'évaluateur qui va effectuer l'inspection de la tâche. En revanche, la quatrième colonne lui permet de choisir la méthode d'évaluation à utiliser. Au stade actuel des travaux, nous avons implémenté la première et la troisième façons de choisir une méthode d'évaluation (voir §3.4.5). Pour la première façon, le superviseur doit choisir une méthode dans la liste à l'exception de celle dénommée « automatique ». Pour la troisième façon, le superviseur doit sélectionner « Automatique » dans la liste, laissant à l'environnement CWE le choix de la méthode d'évaluation à utiliser.



Cet écran permet aussi au superviseur de créer une autre session<sup>4</sup>. On crée une session si on veut faire évaluer une même tâche d'une façon différente, c'est-à-dire avec un autre évaluateur ou avec une autre méthode.

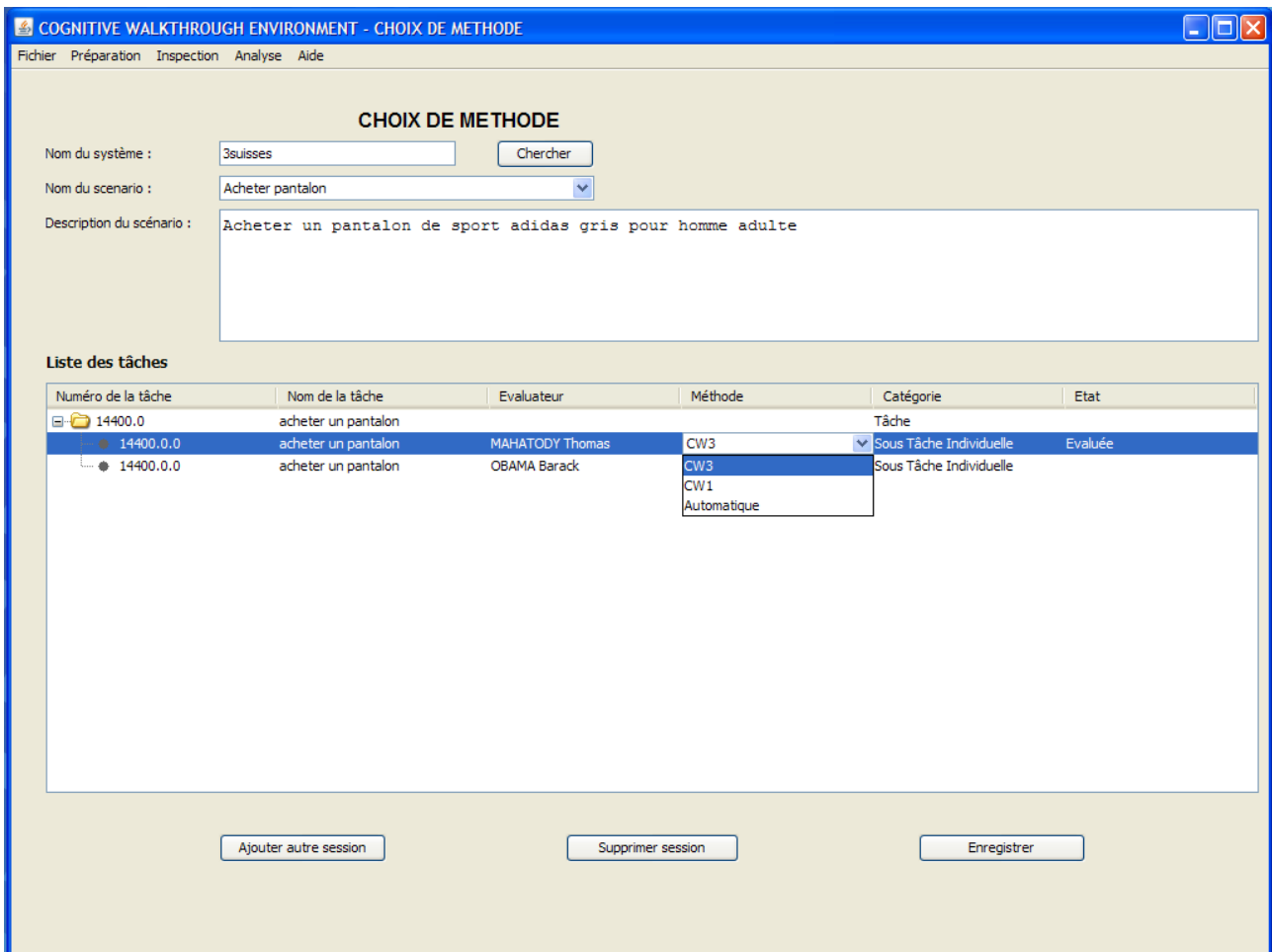


Figure 4.19 : Ecran de choix de méthode

#### 4.3.2.10 Cas d'utilisation : « Inspecter les tâches »

Après la phase de préparation, les évaluateurs peuvent effectuer les inspections. Chacun doit s'identifier avant de commencer l'inspection. Ensuite il doit utiliser l'écran « Ecran d'inspection » (figure 4.20). Ce dernier est obtenu à partir du menu « Analyse » item « Inspection ». Il contient dans la partie haute la fiche d'évaluation (voir annexe A) qui contient le nom de l'évaluateur, la tâche à évaluer, la méthode d'évaluation à utiliser, l'action à évaluer, la question à répondre ainsi que la réponse et la justification de la réponse.

Dans la partie basse, on trouve la fiche problème (voir annexe B) à remplir en cas d'une réponse différente de « OUI ».

<sup>4</sup> Une session est une séance d'évaluation effectuée par un évaluateur avec une méthode d'évaluation choisie.

The screenshot shows a software interface for task inspection. At the top, the title bar reads 'COGNITIVE WALKTHROUGH ENVIRONMENT - ECRAN D'INSPECTION'. Below it, a menu bar contains 'Fichier', 'Préparation', 'Inspection', 'Analyse', and 'Aide'. The main content area is titled 'INSPECTION DE TACHE'. It contains several input fields: 'Evalueateur' with the value 'OBAMA Barack', 'Méthode' with 'CW3', 'Tache' with a dropdown menu showing 'acheter un pantalon', and 'But' with 'Mettre un pantalon de sport adidas gris pour adulte dans le panier'. There are also buttons for 'Terminer', 'Action suivante', and 'Action' with the value 'Cliquer le lien sport'. A 'Question suivante' section is present with a question: 'Est-ce que l'utilisateur tente de réaliser l'effet désiré ?' and a 'Réponse' dropdown menu set to 'Oui'. Below this is a 'Justification' text area. At the bottom, there is a 'Problème d'utilisabilité' section with an 'Ajouter' button and several fields: 'Numéro' (0), 'Description', 'Justification', 'Frequence' (Rarement), 'Sévérité' (Tolérable), 'Pourcentage' (0), 'Catégorie' (Utilisateur), and 'Recommandation'.

Figure 4.20 : Ecran d'inspection de tâche

### 4.3.2.11 Cas d'utilisation : « Calculer les tableaux de bord »

Le superviseur peut analyser les résultats des sessions d'évaluation effectuées par les évaluateurs. Il doit utiliser l'écran « Tableau de bord résultats ». Cet écran est obtenu à partir du menu « Analyse », item « Calculer tableau de bord ». Il contient trois onglets : l'onglet « Problèmes par évaluateur », l'onglet « Synthèse des problèmes par catégorie » et l'onglet « Synthèse des problèmes par évaluateur ».

L'onglet « Problèmes par évaluateur » contient le nom du système, le scénario et la tâche évaluée, le nom de l'évaluateur ainsi que la méthode d'évaluation qu'il a utilisée. La liste des problèmes avec ses différents caractéristiques (gravité, catégorie, etc.) apparait dans un tableau (voir figure 4.21). Pour visualiser la liste des problèmes détectés par tous les évaluateurs ayant évalué une même tâche avec la même méthode, il faut choisir « Tout » dans la liste des évaluateurs.

L'onglet « Synthèse des problèmes par catégorie » donne le nombre de problèmes par catégorie pour la tâche indiquée dans l'onglet « Problème par évaluateur ». Cette synthèse se présente sous forme de tableau dans la partie basse et sous forme graphique dans la partie haute (voir figure 4.22).

L'onglet « Synthèse des problèmes par évaluateur » calcule le nombre de problèmes détectés par chaque évaluateur sur une même tâche avec la même méthode (voir figure 4.23). Il faut donc choisir au préalable « Tout » dans la liste des évaluateurs dans l'onglet « Problèmes par évaluateur ».

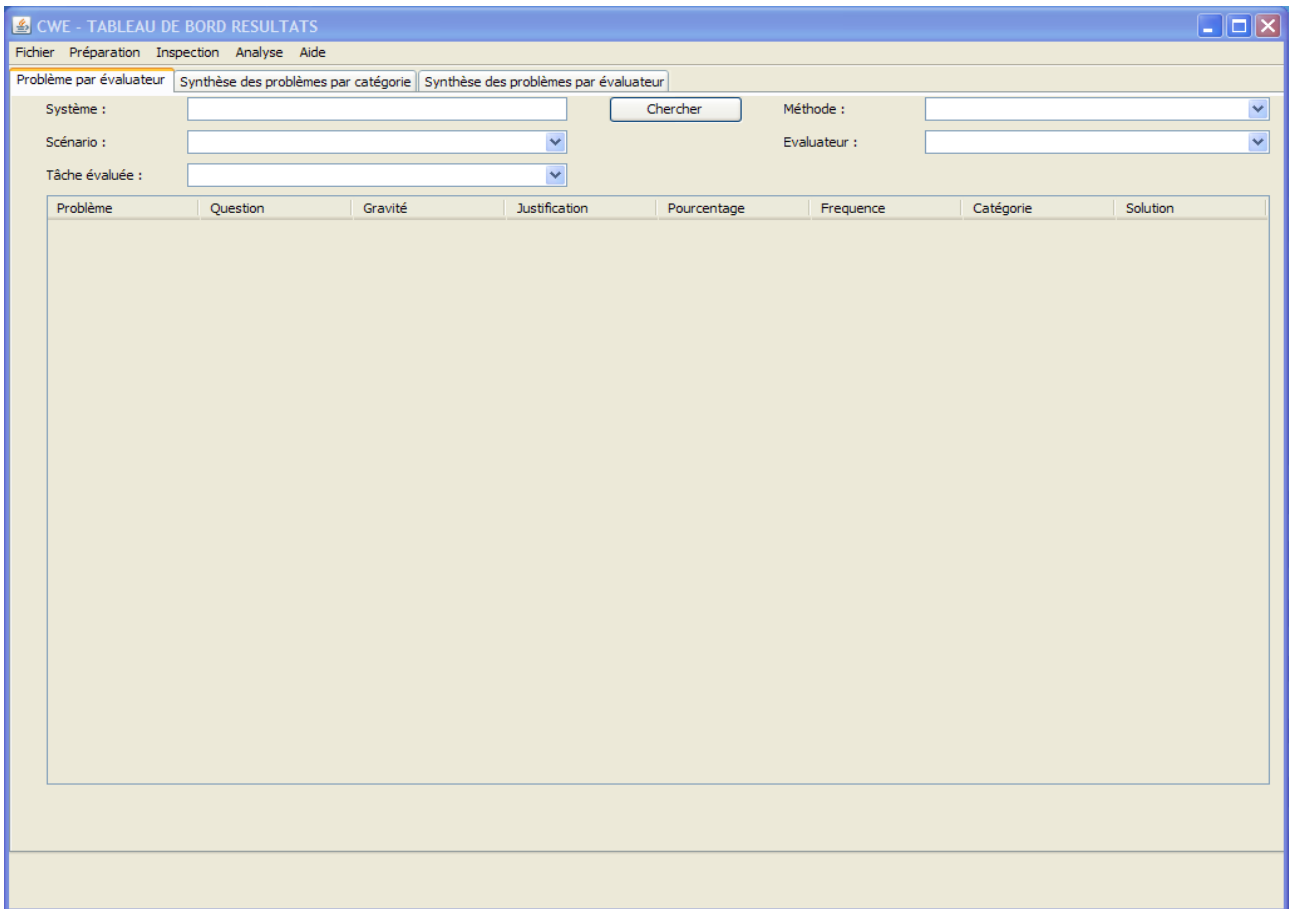


Figure 4.21 : Problèmes par évaluateur

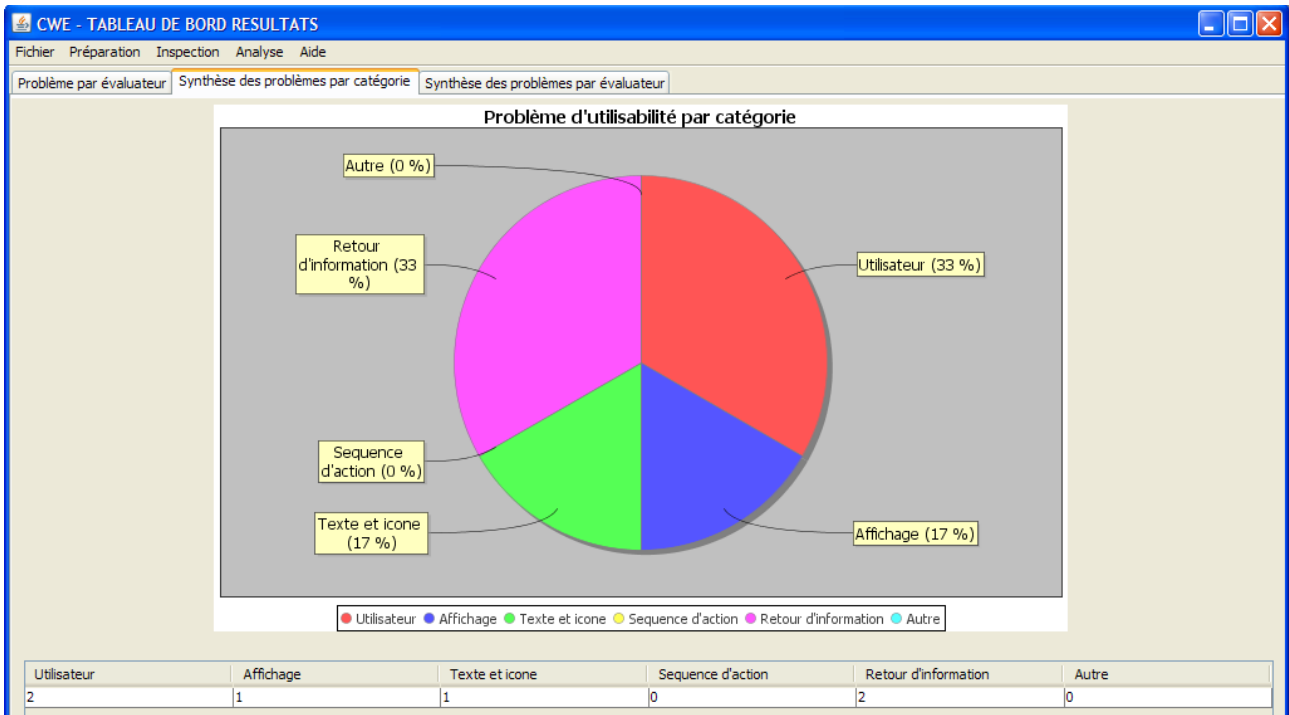


Figure 4.22 : Synthèse des problème par catégorie

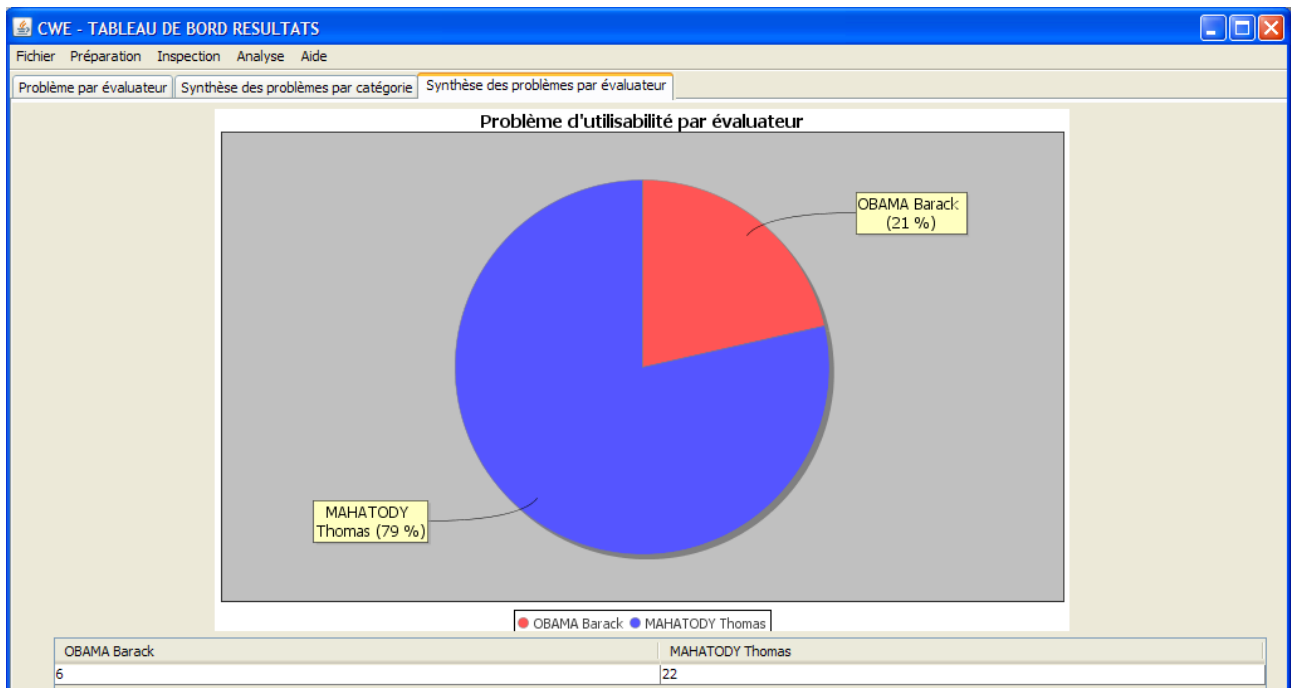


Figure 4.23 : Synthèse des problèmes par évaluateur

### 4.3.3 Vue logique

Dans cette partie, l'organisation de l'environnement CWE en paquetages est décrite, puis on va décrire quelques illustrations de réalisation des cas d'utilisation les plus importants pour montrer l'aspect dynamique de l'environnement.

#### 4.3.3.1 Paquetages de conception architecturale significatifs

Pour la vue logique, nous allons présenter le diagramme de packages (figure 4.24). Il comprend 3 paquetages à savoir : `org.thomah.m2vc.core`, `org.thomah.cwe.orm` et `org.thomah.cwe.view`.

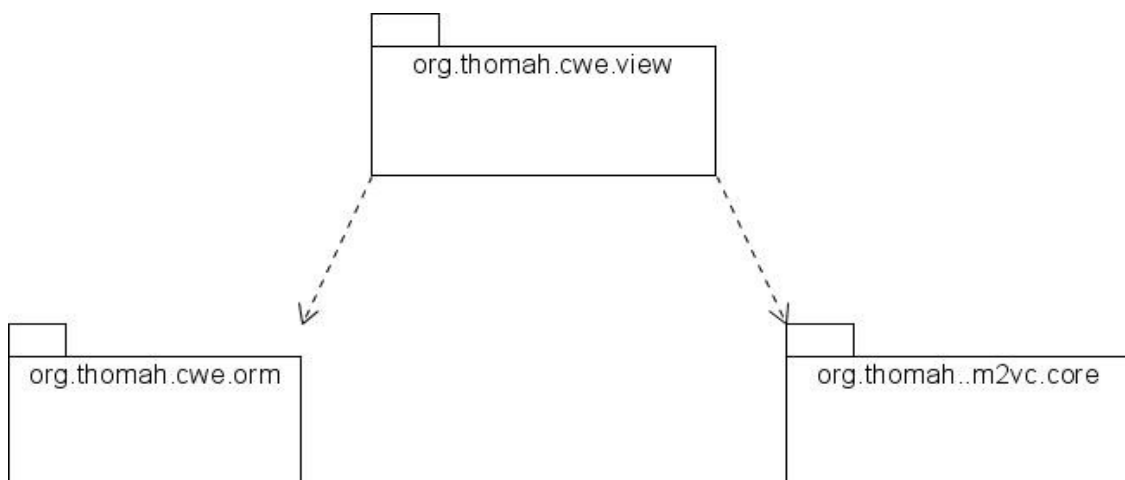


Figure 4.24 : Diagramme de paquetage

Le package org.thomah.m2vc.core constitue un ensemble de service qui prend en charge de l'agencement des IHM. Il comprend entre autre la classe BaseFrame que toutes interfaces doivent héritées (voir figure 4.25).

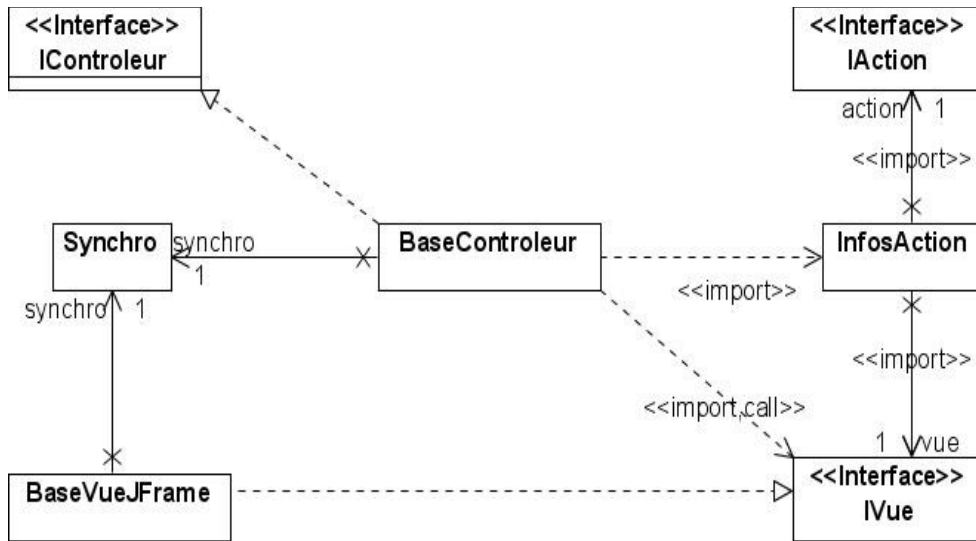


Figure 4.25 : Paquetage org.thomah.m2vc.core

Le paquetage org.thomah.cwe.orm comprend tous les classes persistantes du modèle de domaine (Chapitre 3, figure 3.2), les classes de fabrique de tous les classes persistantes ainsi que la classe CwePersistantManager.

Le paquetage org.thomah.cwe.view constitue l'ensemble des classes permettant de manipuler et utiliser CWE (voir figure 4.26).

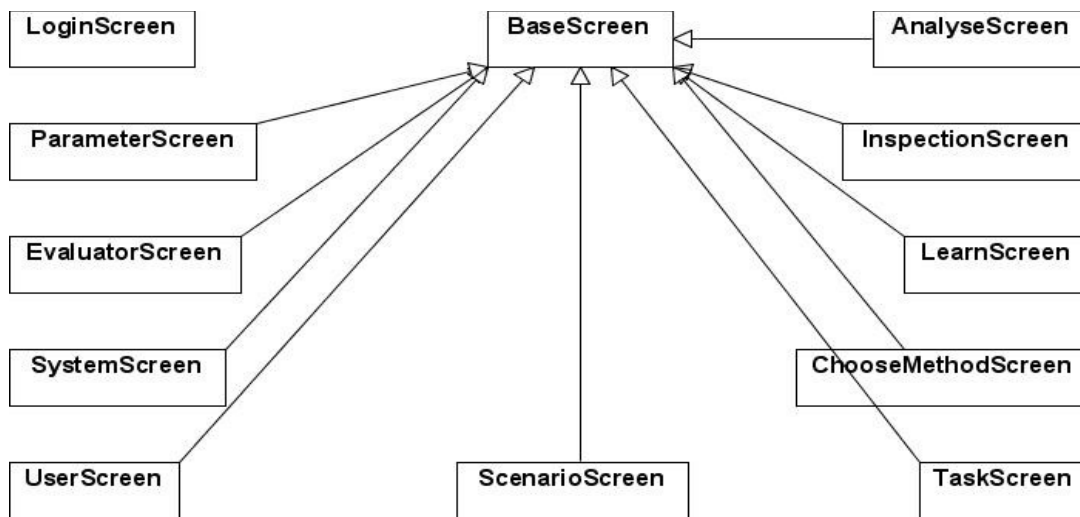


Figure 4.26 : Paquetage org.thomah.cwe.view

Pour faciliter la compréhension de l'architecture de l'environnement CWE, la figure 4.27 montre ce diagramme de paquetage de la figure 4.26 en terme de diagramme de navigation écran.

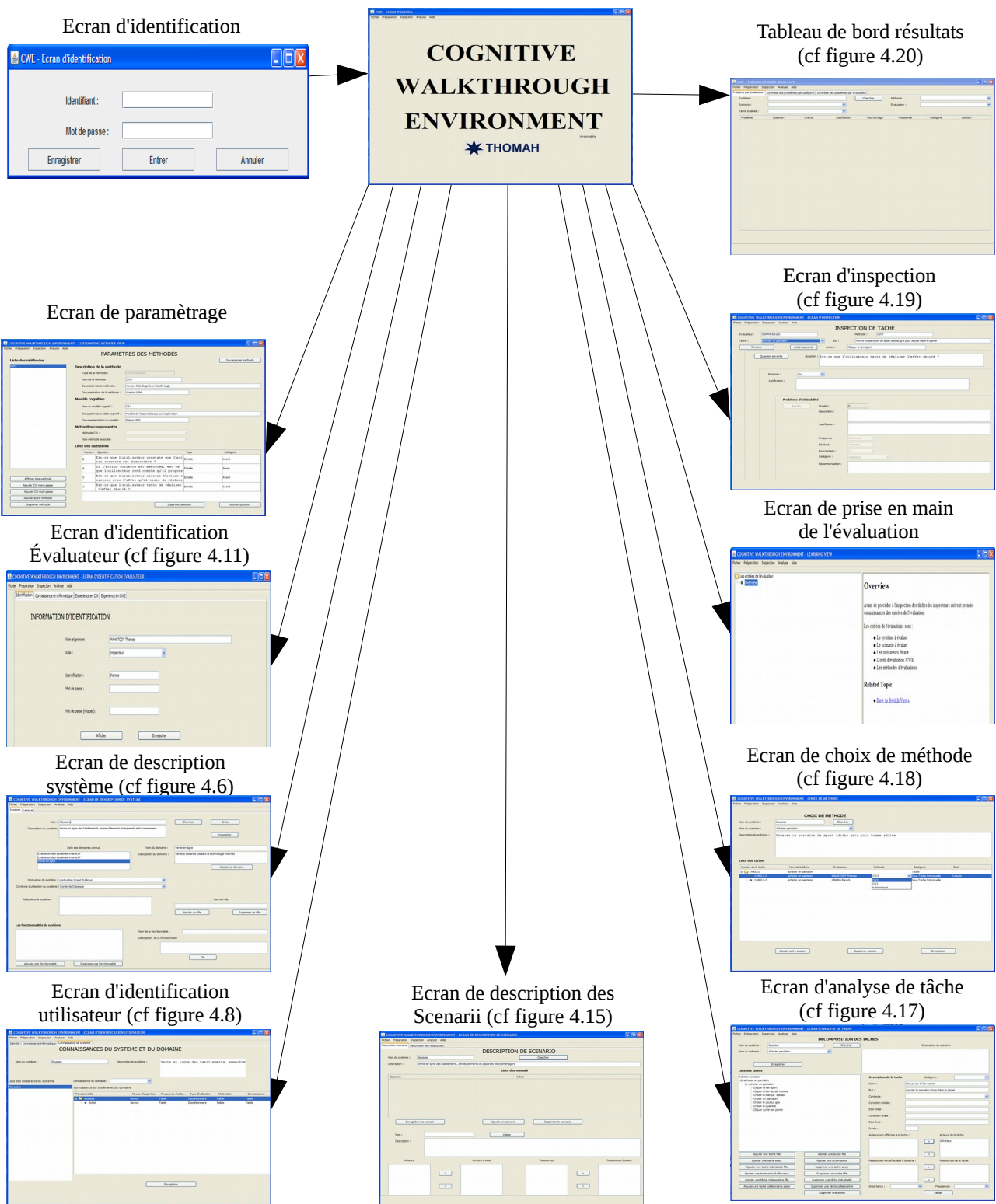


Figure 4.27 : Diagramme de navigation écran

### 4.3.3.2 Illustrations de réalisation des cas d'utilisation

Cette section illustre comment le logiciel travaille en réalisant quelques cas d'utilisation. Pour cela, nous allons évaluer un site web dénommé « 4Belges ».

**Nom du système :** 3Belges

**Description du système :** Vente en ligne des habillements, ameublements et appareils électroménagers.

**Nom du domaine du système :** Vente en ligne

**Description du domaine du système :** Vente à distance utilisant la technologie internet

**Motivation du système :** Grand public

**Contexte d'utilisation du système :** Contexte classique

**Rôles:**

- Acheteur : internautes clients du site
- Visiteur : internautes quelconque qui visitent le site pour de raisons quelconques.

**Fonctionnalités:**

- Naviguer : Naviguer dans le site
- Choisir : choisir un article dans le site
- identifier : Authentifier par mot de passe
- Payer : Payer les achats

**Utilisateur:**

- Ménagère :
- Jeune :
- Personne âgée :

**Scénario : Acheter pantalon**

Une ménagère achète un pantalon de sport pour homme adulte de marque Daddydas

Scénario	Fonctionnalités			
	Naviguer	Choisir	identifier	Payer
Une ménagère achète un pantalon de sport pour homme adulte de marque Daddydas		X		

**Décomposition en tâche**

Tâche : achète un pantalon de sport pour homme adulte de marque Daddydas

Sous-tâche individuelle : achète un pantalon de sport pour homme adulte de marque Daddydas

**Actions :**

- Aller dans le rayon sport
- Aller dans le rayon textile homme
- Aller dans le rayon Daddydas
- Aller dans le rayon pantalon
- Identifier le pantalon gris
- Désigner le nombre de pantalon à commander
- Ajouter le pantalon choisi dans le panier
- Mettre un pantalon de sport Daddydas gris pour adulte dans le panier

**Superviseur :** thomah

**Choix de méthode et évaluateurs :**

Tâche	Evaluateur	Méthode
achète un pantalon de sport pour homme adulte de marque Daddydas	Michel	CW3
	Norah	CW3

**Cas d'utilisation: Description du système à évaluer**

Avant de faire évaluer le site web 3Belges par des inspecteurs, le superviseur doit décrire cette application avec CWE. Pour cela, il ou elle doit utiliser l'écran « Ecran de description de système ». Selon le diagramme de la figure 4.29, on ne peut pas réaliser les autres cas d'utilisation si celui-ci n'est pas réalisé.

**Cas d'utilisation : Identifier les utilisateurs**

Si le cas d'utilisation « Description du système à évaluer » est réalisé, alors on peut réaliser le cas d'utilisation « Identifier les utilisateurs ». Sa réalisation pour le site web 3Belges est décrite dans le diagramme de la figure 4.30.

**Cas d'utilisation : Définir les connaissances informatiques des utilisateurs**

Si le cas d'utilisation « Identifier les utilisateurs » est réalisé, alors on peut réaliser le cas d'utilisation « Définir les connaissances informatiques des utilisateurs ». Sa réalisation pour le site web 3Belges est décrite dans le diagramme de la figure 4.28.

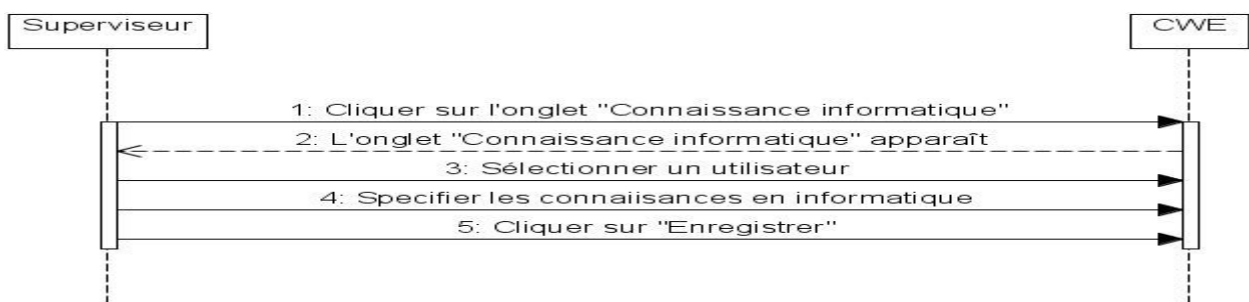


Figure 4.28 : Réalisation du cas d'utilisation « Définir les connaissances informatiques des utilisateurs »



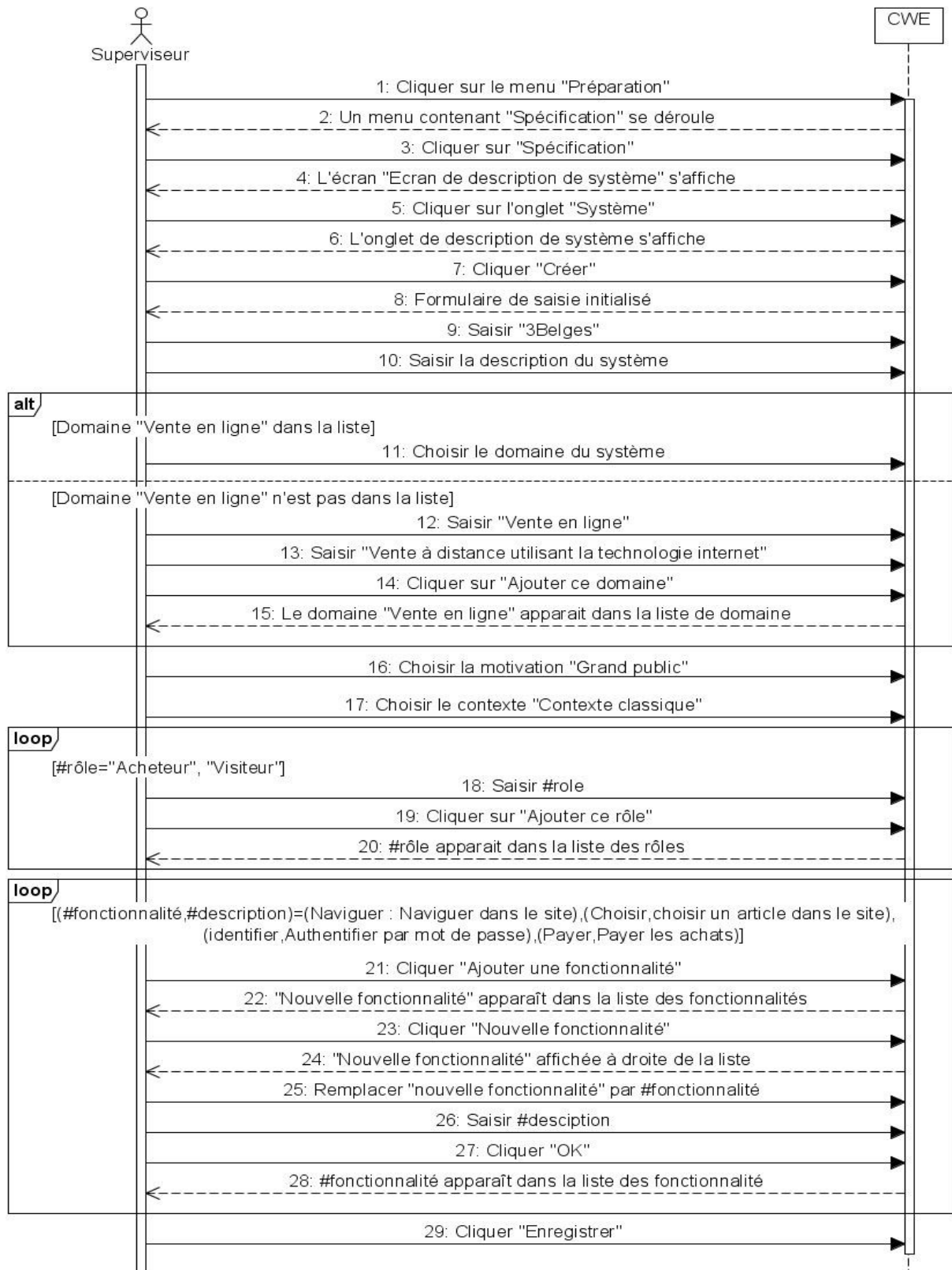


Figure 4.29 : Réalisation du cas d'utilisation « Décrire le système à évaluer »



Figure 4.30 : Réalisation du cas d'utilisation « Identifier les utilisateurs »

**Cas d'utilisation : « Définir les connaissances des utilisateurs concernant le système »**

Comme le cas d'utilisation « Définir les connaissances informatiques des utilisateurs », la réalisation du cas d'utilisation « Définir les connaissances des utilisateurs concernant le système » dépend de la réalisation du cas d'utilisation « Identifier les utilisateurs ». Sa réalisation pour le site web 3Belges est décrite dans le diagramme de la figure 4.31.

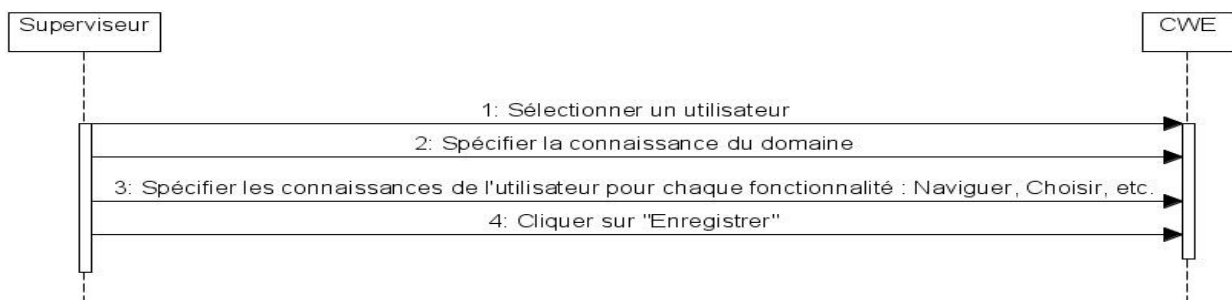


Figure 4.31 : Réalisation du cas d'utilisation « Définir les connaissances des utilisateurs concernant le système »

### Cas d'utilisation : « Décrire les scénarii à évaluer »

Le cas d'utilisation « Décrire les scénarii à évaluer » ne peut être réalisé que si le cas d'utilisation « Décrire le système à évaluer » est réalisé. Sa réalisation pour le site web 3Belges est décrite dans le diagramme de la figure 4.32.

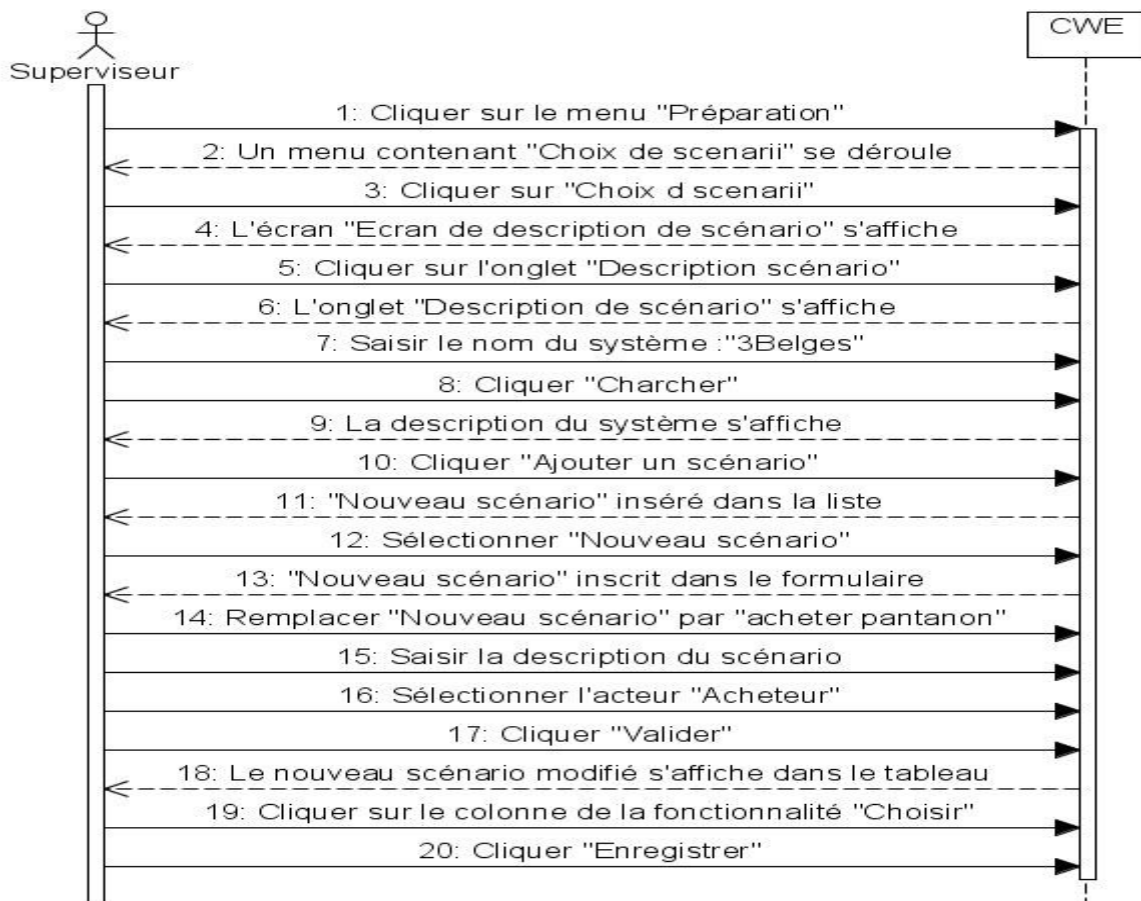


Figure 4.32 : Réalisation du cas d'utilisation : « Choisir les scénarii à évaluer »

### Cas d'utilisation : « Analyser les tâches »

La réalisation du cas d'utilisation « Analyser les tâches » dépend de la réalisation du cas d'utilisation « Décrire les scénarii à évaluer ». En effet chaque scénario décrit doit être décomposé en suite d'action en vue d'une inspection plus tard. Pour le site web 3Belges, la réalisation de ce cas d'utilisation est décrit par le diagramme de la figure 4.33.

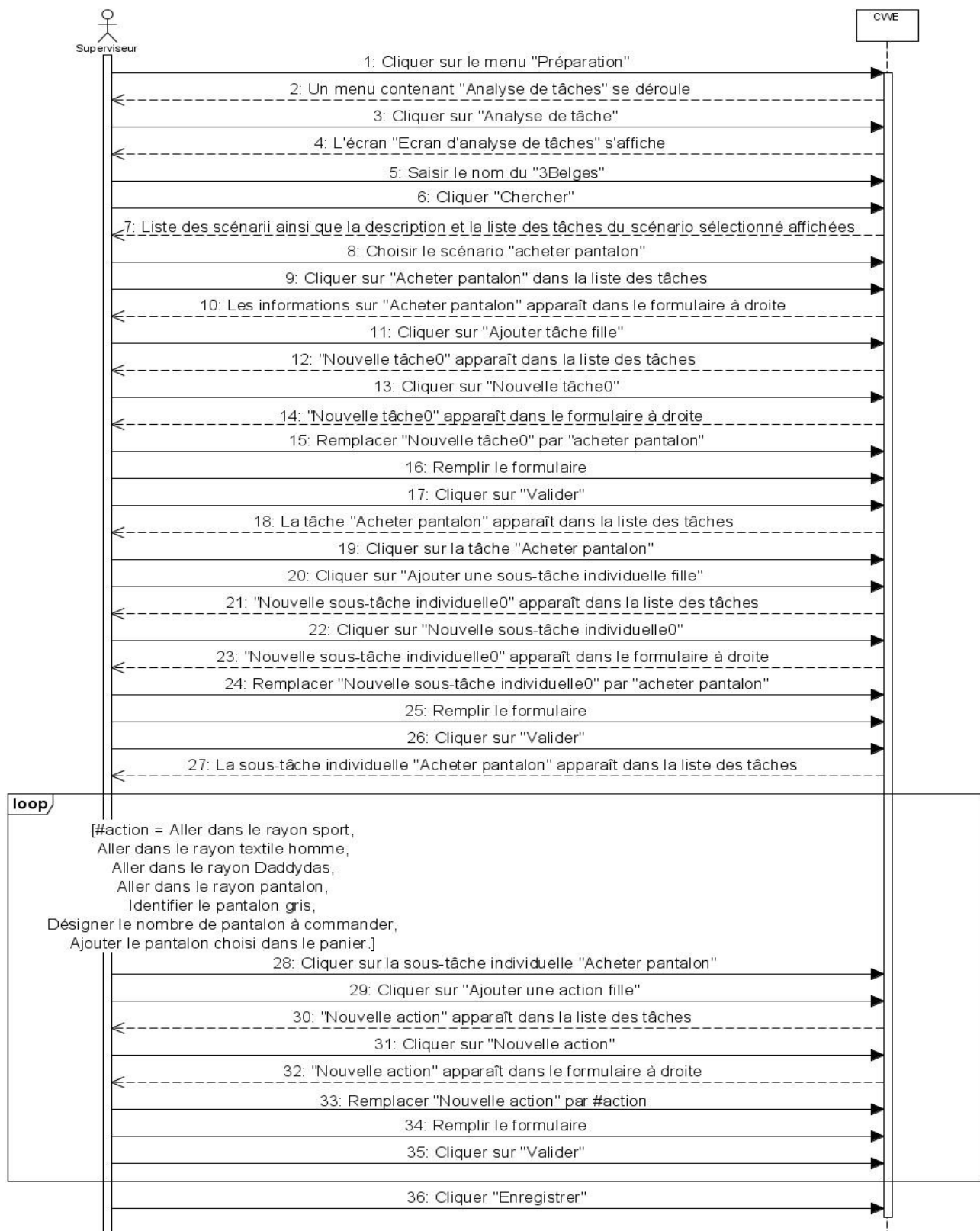


Figure 4.33 : Réalisation du cas d'utilisation : « Analyser les tâches »

### Cas d'utilisation : « Choisir la méthode d'évaluation »

La réalisation du cas d'utilisation « Choisir la méthode d'évaluation » dépend de la réalisation du cas d'utilisation « Analyser les tâches ». En effet, la désignation de méthode à utiliser se fait au niveau des sous-tâches individuelles ou des sous-tâches collaboratives. Durant la réalisation de ce cas d'utilisation, on choisit aussi l'inspecteur qui va effectuer l'inspection. Ainsi, avant de réaliser ce cas d'utilisation, il faut que l'environnement ait des évaluateurs dans sa base de données. Pour le site web 3Belges, la réalisation de ce cas d'utilisation est décrit par le diagramme de la figure 4.34.

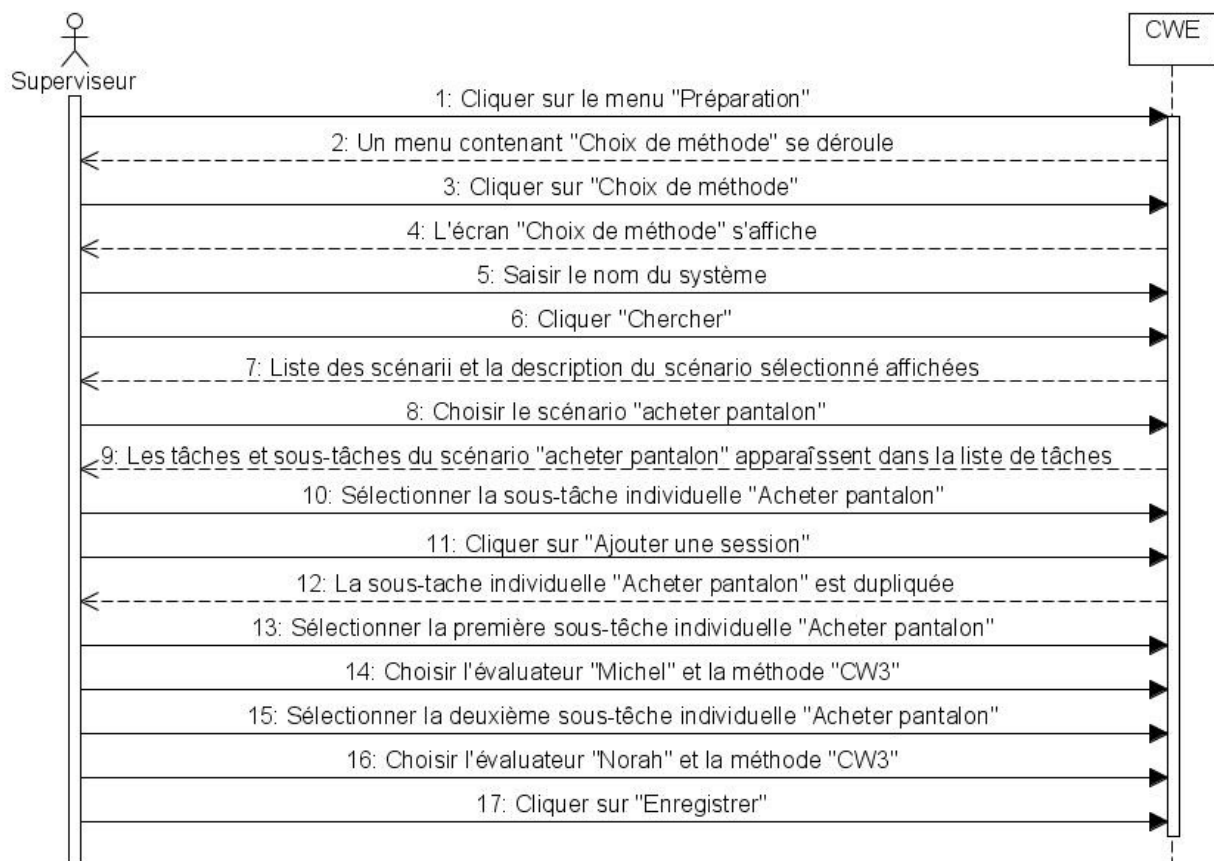


Figure 4.34 : Réalisation du cas d'utilisation : « Choisir la méthode d'évaluation »

### Cas d'utilisation : « Inspecter les tâches »

La réalisation du cas d'utilisation « Inspecter les tâches » dépend de la réalisation du cas d'utilisation « Choisir la méthode d'évaluation ». Si les méthodes d'évaluation sont choisies et les inspecteurs sont désignés, chaque inspecteur peut procéder à l'inspection des tâches qui lui sont attribuées. Il doit s'identifier à l'environnement et dès qu'il entre dans l'écran d'inspection, les tâches qu'il doit inspecter apparaissent automatiquement dans la liste des tâches. Pour le site web 3Belges, la réalisation de ce cas d'utilisation est décrit par le diagramme de la figure 4.35.

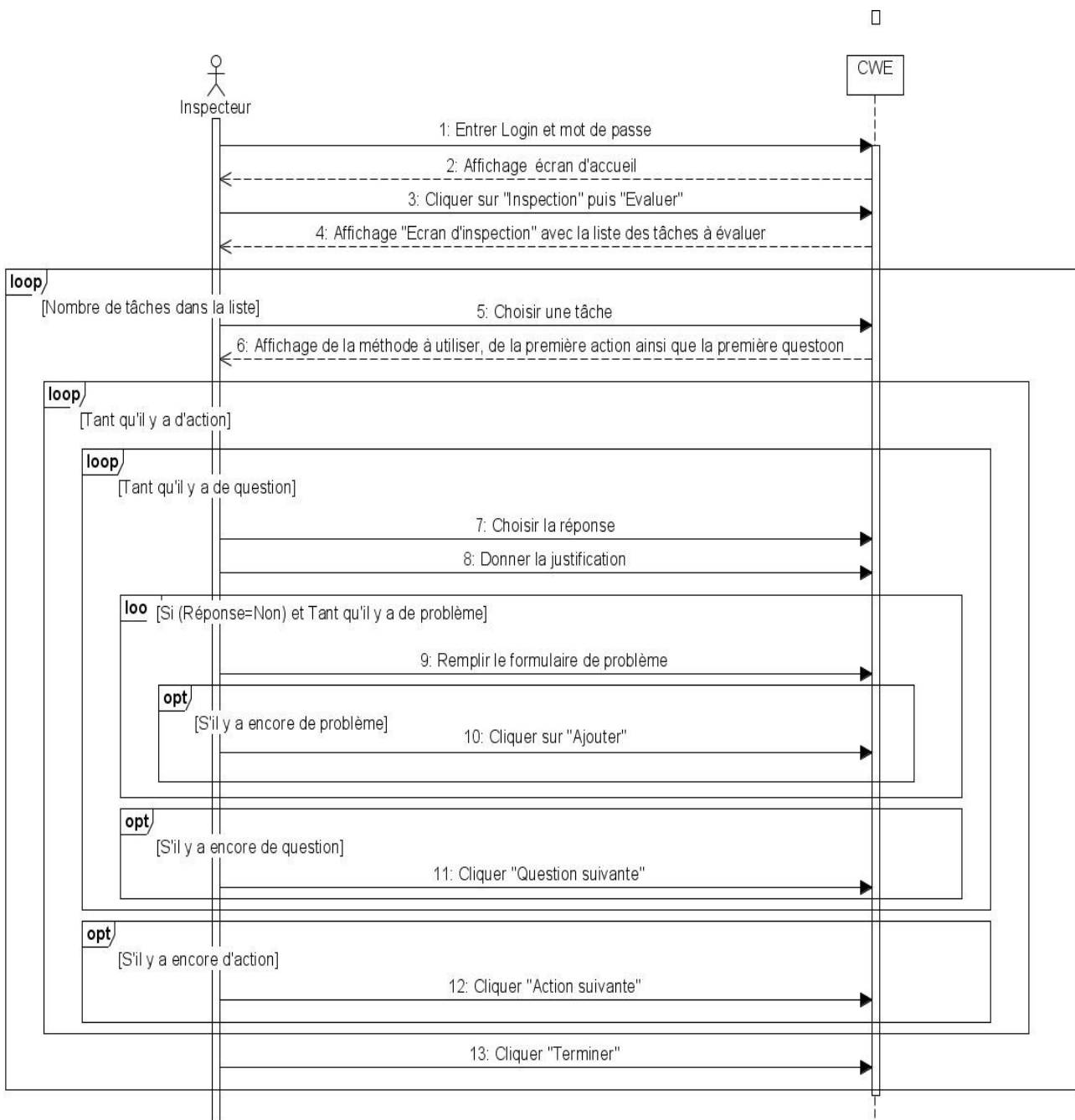


Figure 4.35 : Réalisation du cas d'utilisation : « Inspecter les tâches »

### Cas d'utilisation : «Calculer les tableaux de bord »

Si le cas d'utilisation « Inspecter les tâches » est réalisé, c'est à dire que tous les évaluateurs ont terminé les inspections des tâches, alors on peut réaliser le cas d'utilisation « Calculer les tableaux de bord ». Pour le site web 3Belges, la réalisation de ce cas d'utilisation est décrit par le diagramme de la figure 4.36.

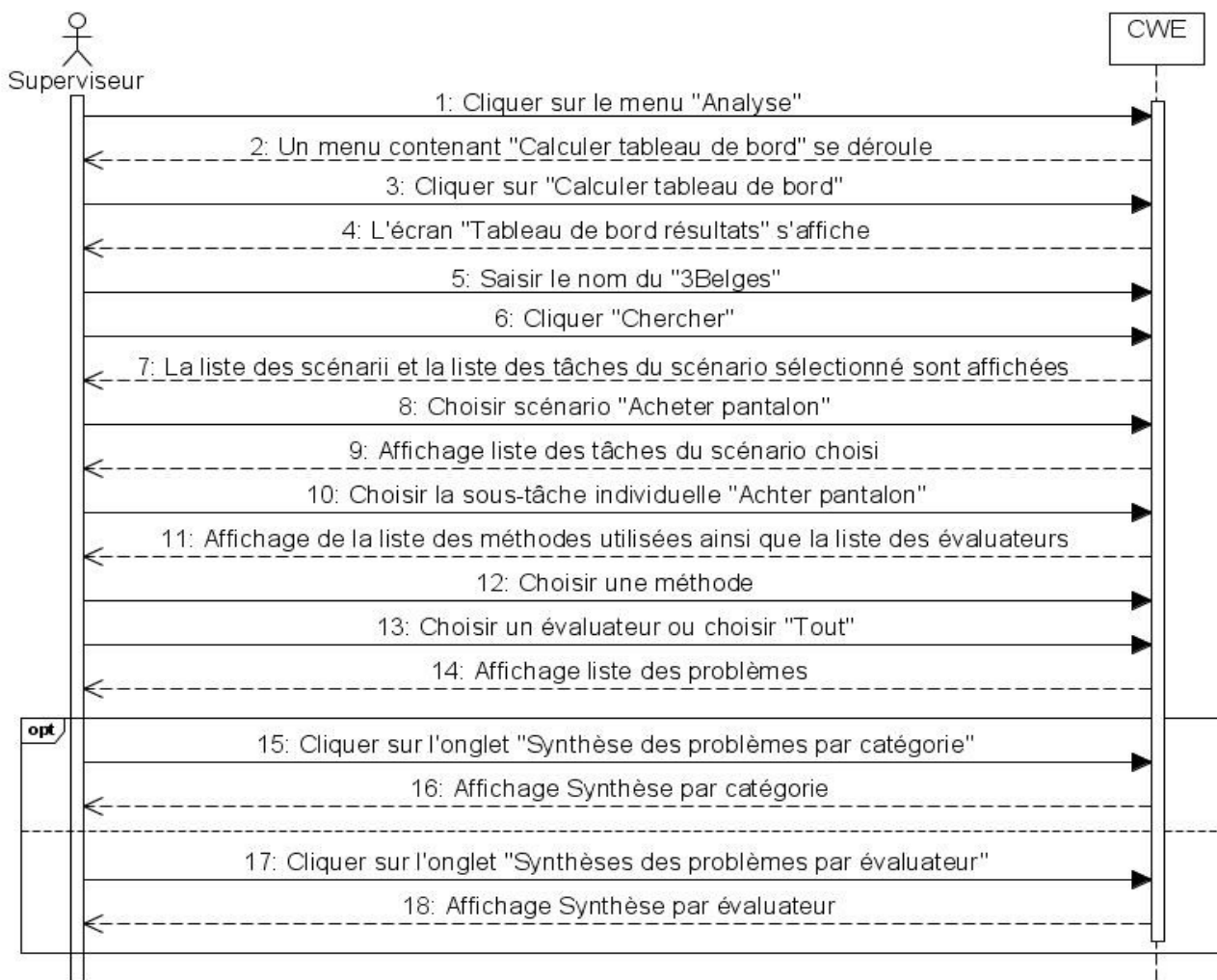


Figure 4.36 : Réalisation du cas d'utilisation : « Calculer les tableaux de bord »

## 4.4 Conclusion

Nous avons présenté dans ce chapitre une maquette de l'environnement CWE. Cette maquette couvre le processus d'évaluation : de la préparation à l'analyse des résultats. C'est une maquette réalisé en Java et qui utilise le système de gestion de base de données MySQL pour mémoriser les données persistantes manipulées par l'environnement. La présentation qu'on a faite montre entre autre l'organisation interne et l'organisation externe de l'environnement CWE à partir des décisions architecturales que nous avons pris.

Concernant l'organisation externe, on a vu les différentes interfaces homme-machine de l'environnement. On a montré aussi comme fonctionne l'environnement en réalisant les cas d'utilisation significatif de l'environnement.

Concernant l'organisation interne, les trois paquetages significatifs ont été décrits. On a montré aussi comment est pris en compte le multilinguisme de l'environnement. On a décrit aussi la façon de réaliser le mécanisme de persistance. Dans le chapitre suivant, nous allons présenter l'évaluation de CWE.

# Chapitre 5 - Evaluation de la première maquette de l'environnement CWE

---

## Plan du chapitre

5.1- Introduction.....	2
5.1.1 - La page de titre.....	2
5.1.2 - Un sommaire exécutif.....	2
5.1.3 - Une introduction.....	3
5.1.3.1 - Description complète du produit.....	3
5.1.3.2 - Objectifs de l'évaluation.....	3
5.1.4 - La méthodologie.....	3
5.1.5 - Résultats.....	4
5.1.5.1 - Analyse des données.....	4
5.1.5.2 - Présentation des résultats.....	4
5.2- Sommaire exécutif.....	4
5.3- Présentation de l'évaluation.....	4
5.3.1 - Description complète du produit à évaluer.....	4
5.3.2 - Objectifs de l'évaluation.....	5
5.4- Méthode.....	6
5.4.1 - Participants.....	6
5.4.2 - Contexte de produit évalué.....	8
5.4.2.1 - Tâches.....	8
5.4.2.2 - Environnement d'évaluation.....	8
5.4.2.3 - Configuration des ordinateurs des participants.....	9
5.4.3 - Stratégie de l'évaluation.....	9
5.4.3.1 - Procédure d'évaluation.....	9
5.4.3.2 - Instructions générales aux participants.....	10
5.4.3.3 - Instructions concernant les tâches des participants.....	10
5.4.4 - Métriques utilisées.....	10
5.5- Résultats de l'évaluation.....	11
5.5.1 - Résultats des tests fonctionnels.....	11
5.5.2 - Résultats de Cognitive Walkthrough version 3.....	13
5.5.3 - Discussion sur les deux évaluations.....	15
5.6- Conclusion.....	16

---



## 5.1 Introduction

On a vu dans le chapitre 1 que l'évaluation est une étape importante dans le processus de conception et de développement d'un système. Ce chapitre présente les évaluations que nous avons entreprises dans le cadre du processus de conception et de développement de CWE.

Nous adoptons le format Common Industry Format for Usability Test Report v2.0 (CIF 2.0) [Scholtz & Morse, 2002] pour la présentation des résultats de cet évaluation. Ce format facilite les échanges entre consultants mais aussi entre consultants et universitaires sur les données d'évaluation utilisateurs et afin que les consommateurs soient mieux informés sur la qualité ergonomique des logiciels interactifs. Ce format précise notamment les points qui doivent être abordés dans les rapports d'évaluation à savoir : les tâches proposées aux utilisateurs, l'environnement d'évaluation (matériel et logiciel), le protocole d'évaluation, les méthodes de recueil de données, les techniques d'analyse mise en œuvre et les résultats de l'évaluation. Dans un premier, nous allons présenter brièvement la structure d'un rapport au format CIF 2,0. Puis dans la deuxième partie de chapitre nous présentons l'évaluation que nous avons faite. Et enfin, nous terminons ce chapitre par une conclusion.

Ainsi nous rappelons qu'un rapport d'évaluation au format CIF 2.0 comporte les cinq sections suivantes :

### 5.1.1 La page de titre

Dans la page de titre doivent être indiqués :

- le format de rapport doit être indiqué à savoir : Common Industry Format for Usability Test Report v2.0 ainsi que l'information de contact,
- Le nom du produit et la version à tester,
- le nom de la personne qui dirige l'évaluation,
- la date de l'évaluation,
- la date de préparation du rapport,
- le nom de la personne qui prépare le rapport,
- le nom de la compagnie cliente,
- le nom de la personne de contact de la compagnie cliente,
- le (ou les) nom(s) de (ou des) personnes à contacter pour des questions et/ou des clarifications,
- numéro de téléphone du fournisseur du produit à évaluer,
- adresse électronique du fournisseur du produit à évaluer,
- adresse postale du fournisseur du produit à évaluer.

### 5.1.2 Un sommaire exécutif

Cette section fournit un aperçu de haut niveau de l'évaluation. L'objectif de cette section est de fournir des informations pour le décideur du service d'approvisionnement dans les organisations

clientes. Ces personnes ne peuvent pas lire le corps technique de ce document. Cette section commence sur une nouvelle page et se termine par un saut de page pour faciliter son utilisation comme un résumé autonome.

## 5.1.3 Une introduction

Nous appellerons cette section plus bas « Présentation de l'évaluation ». Cette section comporte deux sous-section :

### 5.1.3.1 Description complète du produit

Dans cette sous-section, les renseignements suivants doivent être fournis :

- Le nom et la version du produit,
- Les parties du produit qui ont été évaluées,
- La population d'utilisateurs pour laquelle le produit est destiné,
- Une brève description de l'environnement dans lequel il doit être utilisé,
- Le type de travail de l'utilisateur qui est supporté par le produit.

### 5.1.3.2 Objectifs de l'évaluation

Et pour cette deuxième section, les renseignements suivants doivent être fournis :

- Les objectifs de l'évaluation et les domaines d'intérêt spécifiques. Les objectifs peuvent être l'évaluation de la performance des tâches et la satisfaction subjective de l'utilisation du produit.
- Il faut mentionner ici aussi les raisons pour lesquelles l'évaluation se focalisait sur un sous-ensemble du produit, si l'ensemble du produit n'a pas été évalué.

## 5.1.4 La méthodologie

Cette section est décrite la méthodologie d'évaluation. Des informations suffisantes doivent être fournies pour permettre aux évaluateurs indépendants de reproduire la procédure utilisée dans l'évaluation. Ces informations concernent :

- Les participants,
- Le contexte d'utilisation du produit dans l'évaluation,
- La conception de l'évaluation,
- Les métriques d'utilisabilité,
- La liste des tâches,
- L'environnement de test,
- Les configurations des outils utilisés,
- Les documents d'évaluation comprenant les procédures et instructions de l'évaluation.

## 5.1.5 Résultats

Cette section comporte deux sous-sections

### 5.1.5.1 Analyse des données

Dans cette sous-section sont décrites les données résultantes de l'évaluation. Il faut y décrire aussi les traitements effectués sur les données et éventuellement les traitements statistiques.

### 5.1.5.2 Présentation des résultats.

Les résultats doivent être présentés sous forme de tableaux et de graphiques.

## 5.2 Sommaire exécutif

Cette section fournit un aperçu de haut niveau de l'évaluation .

Comme décrit précédemment, le produit à évaluer est le premier prototype de l'environnement CWE décrit dans le chapitre 4. Les fonctionnalités à évaluer sont les fonctionnalités significatives de l'environnement CWE. En ce qui concerne les futurs utilisateurs de CWE, on rappelle que d'après le chapitre 3, on les appelle évaluateurs ; ils sont des concepteurs ou des développeurs des systèmes interactifs. Ils peuvent aussi être des ergonomes ou des spécialistes de l'utilisabilité. Et enfin, ils peuvent aussi être les utilisateurs finaux du système à évaluer.

## 5.3 Présentation de l'évaluation

### 5.3.1 Description complète du produit à évaluer

Le produit à évaluer est un environnement d'aide à l'évaluation appelé « Cognitive Walkthrough Environment » (CWE). Il s'agit de la première version de ce produit. Cette version de CWE est conforme à la spécification décrite dans le chapitre 3. On rappelle qu'elle possède entre autre les fonctionnalités suivantes :

- identification de l'évaluateur
- spécification du système à évaluer
- modélisation de l'utilisateur
- choix de scénarii à évaluer
- analyse des tâches
- choix de méthode d'évaluation
- inspection des tâches
- analyse des résultats

Les utilisateurs finaux du logiciel peuvent être les développeurs ou les concepteurs qui ont réalisé le système interactif. Mais ils peuvent être aussi des utilisateurs finaux du système. Et enfin ils peuvent être aussi des ergonomes ou spécialistes de l'utilisabilité.

Le logiciel peut fonctionner avec n'importe quel ordinateur qui possède une machine virtuelle Java (version supérieure ou égale à v1.4.2) et un serveur de base de données MySQL (version supérieure ou égale à 5.1.30).

### 5.3.2 Objectifs de l'évaluation

On peut évaluer quatre types d'information : les programmes et les données, les documents utilisateurs, les documents techniques et les codes sources. Et pour chaque information, on peut évaluer six caractéristiques logicielles :

**Capacité fonctionnelle**

Ensemble d'attributs portant sur l'existence d'un ensemble de fonctions et leurs propriétés données. Les fonctions sont celles qui satisfont aux besoins exprimés ou implicites.

**Fiabilité**

Ensemble d'attributs portant sur l'aptitude du logiciel à maintenir son niveau de service dans des conditions précises et pendant une période déterminée.

**Facilité d'utilisation**

Ensemble d'attributs portant sur l'effort nécessaire pour l'utilisation et sur l'évaluation individuelle de cette utilisation par un ensemble défini ou implicite d'utilisateurs.

**Rendement**

Ensemble d'attributs portant sur le rapport existant entre le niveau de service d'un logiciel et la quantité de ressources utilisées, dans des conditions déterminées.

**Maintenabilité**

Ensemble d'attributs portant sur l'effort nécessaire pour faire des modifications données.

**Portabilité**

Ensemble d'attributs portant sur l'aptitude de logiciel à être transféré d'un environnement à l'autre.

Au stade actuel des travaux les informations à évaluer sont les programmes et les données (cf. tableau 5.1). Les autres informations pourront faire l'objet d'une autre session d'évaluation. Pour cette analyse préliminaire du prototype de l'environnement CWE, les objectifs sont double. Le premier consiste à vérifier si la capacité fonctionnelle de l'environnement est conforme aux spécifications décrites dans le chapitre 3. Le deuxième objectif consiste à évaluer la facilité d'utilisation de l'environnement. Les autres caractéristiques pourront aussi faire l'objet d'une évaluation ultérieure.

Tableau 5.1 : Information et caractéristiques retenues

Information	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Programmes et données	X		X			
Document utilisateur						
Document technique						
Code source						

Pour l'évaluation concernant la facilité d'utilisation du logiciel, nous avons restreint l'évaluation à la partie du logiciel concernant la « spécification du système à évaluer ». En effet, l'évaluation de chaque tâche dure environ trois heures. Ainsi, pour évaluer les onze tâches, chaque expert aurait dû consacrer environ trente heures à l'évaluation ; ce qui n'aurait pas été possible dans la mesure où nous avons prévu que chaque expert dispose de trois heures pour effectuer l'évaluation.

## 5.4 Méthode

Cette section fournit des informations suffisantes pour permettre aux évaluateurs indépendant de reproduire la procédure utilisée dans l'évaluation.

- Les participants,
- le contexte d'utilisation du produit dans l'évaluation,
- la liste des tâches,
- l'environnement de test,
- les configurations des outils utilisés,
- les documents d'évaluation comprenant les procédures et instructions de l'évaluation,
- les paramètres d'utilisabilité à mesurer (efficacité, satisfaction, etc.).

### 5.4.1 Participants

Cette section décrit les évaluateurs qui ont participé à l'évaluation en termes de démographie, d'expérience professionnelle, d'expérience en informatique.

Comme on a deux types d'évaluation à réaliser (évaluation des capacités fonctionnelles et évaluation de l'utilisabilité), nous avons aussi deux groupes d'intervenants. Les intervenants du premier groupe (cf. Tableau 5.2) évaluent la capacité fonctionnelle de l'environnement CWE avec la méthode de test fonctionnel. Ils sont au nombre de seize et sont tous des étudiants en master 1 informatique. Ils ont eu un module d'Interaction Homme-Machine durant le semestre. Ils ont une moyenne d'âge de 23 ans.

Les trois intervenants du deuxième groupe évaluent l'utilisabilité de l'environnement CWE avec la troisième version de la méthode Cognitive Walkthrough. Ce sont des experts d'une moyenne d'âge de 47 ans, ayant des connaissances profondes et de nombreuses expériences dans différents domaines, ayant utilisé souvent Cognitive Walkthrough en évaluation des systèmes interactifs.

Tableau 5.2 : Les intervenants du premier groupe d'évaluateurs

	Genre	Age	Education	Occupation/Rôle	Expérience Professionnelle	Expérience en informatique	Expérience sur CWE
E1	Mâle	22	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E2	Mâle	22	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E3	Mâle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E4	Mâle	25	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E5	Femelle	24	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E6	Femelle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E7	Mâle	24	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E8	Mâle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E9	Mâle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E10	Femelle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E11	Femelle	22	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E12	Femelle	25	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E13	Femelle	24	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E14	Mâle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E15	Mâle	23	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
E16	Mâle	21	Bac+4	Evaluateur	1 an	5 ans	Première utilisation
P1	Femelle	44	Bac+8	Evaluateur	10 ans	18	Première utilisation
P2	Mâle	47	Bac+8	Evaluateur	20 ans	31	Première utilisation
P3	Mâle	50	Bac+8	Evaluateur	25 ans	34	Première utilisation

## 5.4.2 Contexte de produit évalué

Cette section décrit les tâches, les scénarios et les conditions sous lesquelles l'évaluation a été réalisée, les fonctionnalités à évaluer, la plate-forme utilisée ainsi que les configurations spécifiques utilisées par les évaluateurs.

### 5.4.2.1 Tâches

Cette sous-section présente la liste des exigences retenues pour l'évaluation. Nous avons retenu comme exigences d'évaluation huit fonctionnalités significatives de l'environnement CWE concernant les trois phases de l'évaluation. Nous avons décidé aussi d'évaluer une exigence non fonctionnelle à savoir la facilité d'utilisation (cf. tableau 5.3).

Tableau 5.3 : Liste des exigences d'évaluation

EX-01	identification de l'évaluateur
EX-02	spécification du système à évaluer
EX-03	modélisation de l'utilisateur
EX-04	choix de scénarii à évaluer
EX-05	analyse des tâches
EX-06	choix de méthode d'évaluation
EX-07	inspection des tâches
EX-08	analyse des résultats
EX-09	Facilité d'utilisation

### 5.4.2.2 Environnement d'évaluation

Cette section se rapporte à la description physique du moyen de l'évaluation. Il doit être décrit ici l'aménagement de l'espace où l'évaluation a eu lieu tel qu'un laboratoire d'utilisabilité, un bureau, etc.). Tous dispositifs ou circonstances appropriés qui pourraient affecter la qualité des résultats, tels que des appareils de contrôle de contrôle visuel et audio, des miroirs à sens unique, ou l'équipement automatique de collecte de données doivent aussi être décrits ici.

Ainsi, pour les évaluateurs du groupe 1, ils ont travaillé dans une salle de travaux pratiques de l'Université de Valenciennes. On a mis à la disposition des évaluateurs des ordinateurs (un par personne), un plan d'évaluation et un cahier d'évaluation (cf. ci-dessous).

Pour les évaluateurs du groupe 2, ils ont travaillé dans un bureau comportant une grande table et un ordinateur portable (Figures 5.2 et 5.3). On leur a fourni un plan d'évaluation, un cahier d'évaluation, des fiches d'évaluation et de description de problèmes (cf. ci-dessous). On rappelle qu'ils ont travaillé à tour de rôle.

### 5.4.2.3 Configuration des ordinateurs des participants

Cette section décrit la configuration des ordinateurs utilisés. Ainsi pour les évaluateurs du groupe 1, chacun d'eux utilise un ordinateur fixe avec un écran de 19 pouces. Dans chaque ordinateur sont installés tous les logiciels nécessaires tels que Java Virtual Machine (version 1.4.2), le SGBD Mysql (version 5.1.30) et CWE. Pour les évaluateurs du groupes 2, ils utilisent un ordinateur portable de 17 pouces où tous les logiciels nécessaires sont installés aussi.

## 5.4.3 Stratégie de l'évaluation

Dans cette section est décrite la conception logique de l'évaluation.

### 5.4.3.1 Procédure d'évaluation

Cette section détaille le protocole d'essai.

On installe les évaluateurs puis on leur distribue les outils de travail :

Le plan d'évaluation (cf. Annexe A) présente l'organisation, l'environnement et la planification des activités d'évaluation de l'environnement CWE. Les fonctionnalités à tester ainsi que les tâches correspondantes sont décrites dans ce document.

Le cahier d'évaluation (cf. Annexes B et C) présente les scénarii à évaluer conformément aux exigences d'évaluation décrites en §5.2.3 L'environnement de l'évaluation et la description de l'environnement CWE y sont présentés aussi. Le cahier d'évaluation de l'annexe B concerne le test fonctionnel, c'est-à-dire celui à destination du groupe 1. Tandis que le cahier d'évaluation de l'annexe C est à destination du groupe 2.

La fiche d'évaluation (cf. Annexe D) est un formulaire d'évaluation basée sur la troisième version de la méthode Cognitive Walkthrough. On y trouve en particulier les quatre questions auxquelles l'évaluateur doit répondre, et ceci pour chaque action décrite dans le cahier d'évaluation.

La fiche problème (cf. figure 5.1) est un formulaire d'enregistrement de problèmes que l'évaluateur doit remplir à la suite d'une réponse négative durant le remplissage de la fiche d'évaluation. Cette fiche permet donc aux évaluateurs de décrire chaque problème qu'il trouve selon notre modèle (cf. §3.3.5).



Référence du scénario :		Date :	
Référence de l'action :		Nom de l'évaluateur :	
Référence de la question :		Numéro du problème :	
<b>Description :</b>			
<b>Justification :</b>			
<b>Fréquence</b>		<b>Sévérité</b>	
<input type="checkbox"/> Rarement <input type="checkbox"/> Souvent <input type="checkbox"/> Très souvent <input type="checkbox"/> Toujours		<input type="checkbox"/> Tolérable <input type="checkbox"/> Modéré <input type="checkbox"/> Sérieux <input type="checkbox"/> Critique	
<b>Pourcentage</b>		<b>Catégorie</b>	
<input type="checkbox"/> 0 <input type="checkbox"/> 25 <input type="checkbox"/> 50 <input type="checkbox"/> 75 <input type="checkbox"/> 100		<input type="checkbox"/> Utilisateur <input type="checkbox"/> Affichage <input type="checkbox"/> Retour d'information <input type="checkbox"/> Séquence d'action <input type="checkbox"/> Texte&Icône <input type="checkbox"/> Autre	
<b>Recommandation :</b>			

Figure 5.1 : Fiche problème (cf. aussi Annexe E)

### 5.4.3.2 Instructions générales aux participants

Après une explication et une lecture du cahier d'évaluation, pour le test fonctionnel, les seize évaluateurs forment huit groupes de deux évaluateurs. Les deux membres de chaque groupe joueront donc les deux rôles : superviseur et inspecteur.

### 5.4.3.3 Instructions concernant les tâches des participants

Les évaluateurs testent l'environnement en procédant à l'évaluation d'un scénario d'un site de réservation de voyage en ligne (déjà étudié dans le cadre de leur module d'enseignement en interaction homme-machine ; les évaluateurs peuvent donc se concentrer sur l'environnement CWE et son évaluation). A chaque cas de test (cf. tableau 5.3) relatif aux exigences de l'évaluation décrits dans §5.2.3, ils sont invités à donner leur avis selon les critères de conformité décrits en §5.2.4.3. La séance a duré trois heures.

## 5.4.4 Métriques utilisées

Pour le test fonctionnel, les critères de succès sont :

- « Conforme » si l'exécution s'est déroulée normalement ;
- « Sous réserve » si l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées ;
- « Non conforme » si l'exécution ne s'est pas déroulée comme prévu.

Pour Cognitive Walkthrough, les critères de succès sont :

- « Conforme » si l'IHM ne présente pas de problèmes d'utilisabilité (ceci correspond à la réponse « OUI » aux questions figurant dans la fiche d'évaluation).

- « Non conforme » si l'IHM présente des problèmes d'utilisabilité (ceci correspond à la réponse « NON » aux questions figurant dans la fiche d'évaluation).

## 5.5 Résultats de l'évaluation

Dans cette section, nous allons décrire puis analyser les résultats des deux types d'évaluation. Ensuite, nous allons relier les deux résultats en vue d'en tirer des conclusions pour l'environnement CWE dans sa globalité.

### 5.5.1 Résultats des tests fonctionnels

Les résultats des tests fonctionnels effectués par les huit groupes sont synthétisés dans le tableau 5.4. On constate tout de suite que seul un groupe sur huit arrive jusqu'à la tâche CT-FONC-08. Les autres groupes ont eu de problème de configuration. Le groupe 1 a pu tester l'ensemble des cas de test de l'environnement CWE avec plus de 70% d'avis « conforme » et moins de 30% d'avis « sous réserve » (cf. figure 5.2). Ce qui veut dire que notre maquette de l'environnement CWE supporte le processus d'évaluation par la méthode Cognitive Walkthrough malgré les erreurs rencontrées par les huit groupes d'évaluateurs. L'analyse des anomalies et commentaires dans les cahiers d'évaluation montre que les erreurs sont liées à l'utilisation de l'environnement. On a pu par exemple relever les anomalies et commentaires suivants:

- « *Certains termes peuvent être incompris par l'utilisateur (discrétionnaire par exemple) »*
- « *Action 16 : le bouton enregistrer est situé tout en haut »*
- « *Le contexte d'utilisation devrait être expliqué (on ne comprend pas bien ce paramètre) »*

La figure 5.3 montre la répartition des groupes d'évaluateur selon les jugements qu'ils ont émis pour chaque cas de test. Ce tableau permet de prioriser les cas de test selon les jugements les concernant. Le cas de test numéro 2 par exemple n'a reçu aucun avis « conforme », ce qui nous a amené à le choisir pour être évalué par les trois experts avec la méthode Cognitive Walkthrough.

Tableau 5.4 : Résultat des tests fonctionnels

	CAS DE TEST								Nombre de cas jugés : <i>Conforme</i>	Nombre de cas jugés : <i>Sous Réserve</i>	Nombre de cas jugés : <i>Non Conforme</i>	Nombre de cas sans jugement
	O1	O2	O3	O4	O5	O6	O7	O8				
Groupe 1	C	S	C	C	S	C	C	C	6	2	0	0
Groupe-2	N	S	S	N	N	-	-	-	0	2	3	3
Groupe 3	S	S	C	S	-	-	-	-	1	3	0	4
Groupe 4	S	S	C	S	S	-	-	-	1	4	0	3
Groupe 5	S	S	S	-	-	-	-	-	0	3	0	5
Groupe 6	S	S	C	S	N	-	-	-	1	3	1	3
Groupe 7	S	S	N	S	C	-	-	-	1	3	1	3
Groupe 8	S	N	S	S	S	-	-	-	0	4	1	3
Nombre de groupes ayant jugé : <i>Conforme</i>	1	0	4	1	1	1	1	1				
Nombre de groupe ayant jugé : <i>Sous Réserve</i>	6	7	3	5	3	0	0	0				
Nombre de groupe ayant a jugé : <i>Non Conforme</i>	1	1	1	1	2	0	0	0				
Nombre de groupes sans avis	0	0	0	1	2	7	7	7				

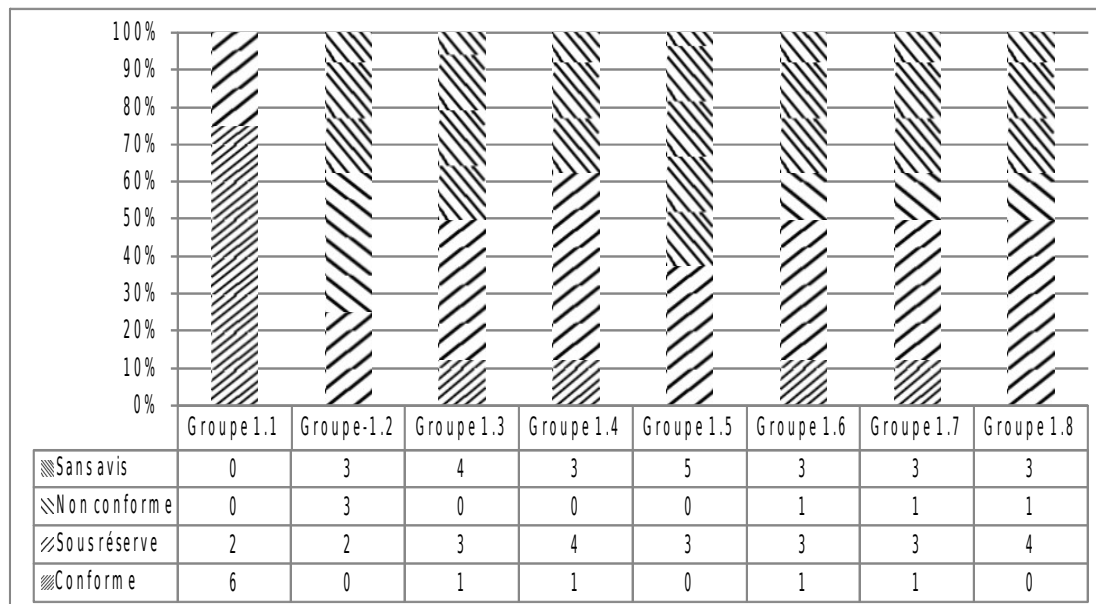


Figure 5.2 : Répartition des tâches selon les avis émis par chaque groupe

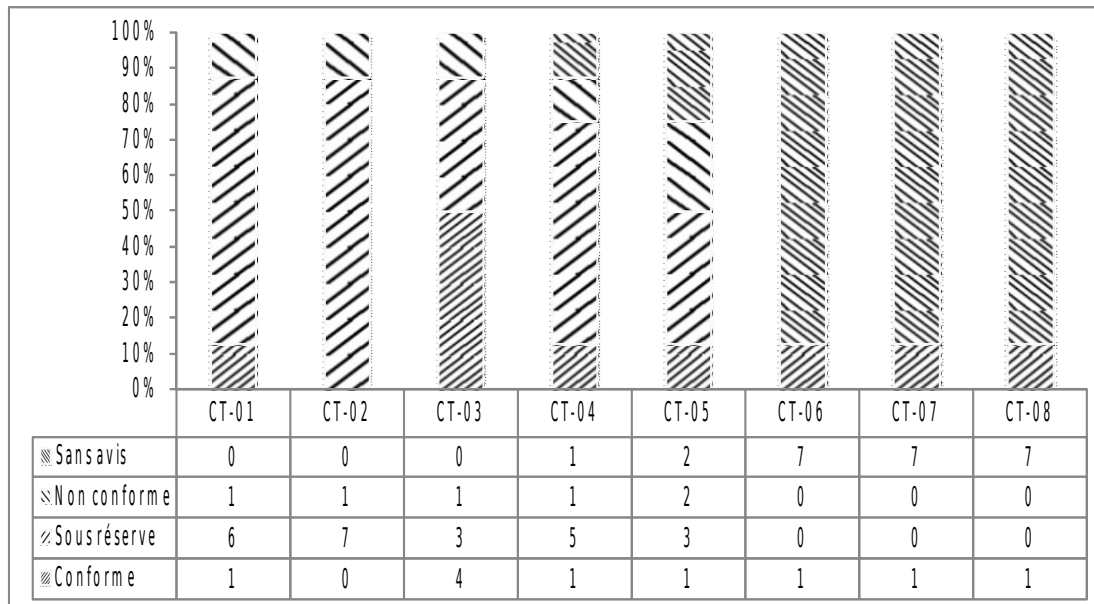


Figure 5.3 : Répartition des groupes d'évaluation selon les avis qu'ils ont émis.

### 5.5.2 Résultats de Cognitive Walkthrough version 3

Les nombres de problèmes trouvés par les experts sont montrés dans le tableau 5.5. Ces problèmes ne sont pas exhaustifs. Dans certaines actions, ils ont trouvé les mêmes problèmes.

Tableau 5.5 : Nombre de problèmes trouvés par chaque expert

	Expert 1	Expert 2	Expert 3
Nombre de problèmes trouvés par expert	22	15	11

La figure 5.6 montre l'intersection entre les sources des problèmes trouvés par les experts. La source du problème est la référence de l'action ainsi que la référence de la question d'où le problème est trouvé. On constate qu'ils ont seulement deux sources en commun. Ces sources sont : action 4/question 4 (cf. tableau 5.6) d'une part, et action 17/question 4 (cf. tableau 5.7) d'autre part. D'après les deux tableaux, on voit qu'il s'agit de problèmes du même type : ce sont des problèmes de guidage.

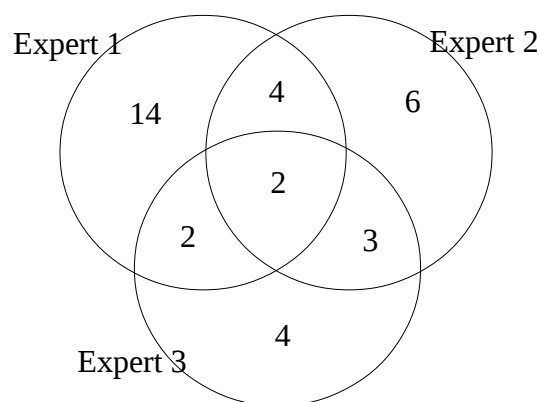


Figure 5.4 : Intersection des sources des problèmes détectés par les experts

Tableau 5.6 : Comparaison des problèmes issus de l'action numéro 4 et de la question numéro 4

	<b>Expert 1</b>	<b>Expert 2</b>	<b>Expert 3</b>
<b>Fréquence</b>	Souvent	Toujours	Toujours
<b>Sévérité</b>	Tolérable	Critique	Modéré
<b>Pourcentage</b>	25	100	100
<b>Catégorie</b>	Texte/Icône	Retour d'information	Retour d'information
<b>Description</b>	« Après avoir cliqué sur créer, on voit qu'il se passe quelque chose, c'est très rapide et la situation reste un peu floue »	« Quand je clique sur créer, il n'y a aucun retour d'informations qui me montre un changement sur la page et qui efface les informations précédentes »	« Ce n'est pas demandé de remplir »
<b>Justification</b>	« des éléments sont apparus, mais pas dans toute les zones et comme tout est de la même couleur, on ne sait pas si c'est important »	« je clique plusieurs fois sur créer et je ne constate pas un changement pour pouvoir réaliser la prochaine action »	
<b>Recommandation</b>	« mettre mieux en évidence peut être par des couleurs les changements »	« Ajouter une indication de changement d'état aussi bien sur le bouton créer que sur la page. Par exemple effacer l'information précédente »	

Le tableau 5.8 montre la répartition des problèmes trouvés par les experts par catégorie. Il montre qu'à l'image de deux problèmes communs trouvés par les experts, la plupart des problèmes trouvés sont liés au problème de guidage : problèmes d'affichages (13), problèmes de retour d'information (8), etc.

Parmi les problèmes de guidage évoqués par les experts, ils ont mis en évidence deux types de problèmes : les problèmes d'organisation d'écran et les problèmes des messages non compris. Les experts 1 et 2 recommandent par exemple de déplacer le bouton « Enregistrer » vers le bas, mettre le bouton « Enregistrer » en haut évoque une mauvaise organisation de l'écran car ils pensent qu'on enregistre toujours à la fin de la page.

Tableau 5.7 : Comparaison des problèmes issus de l'action numéro 17 et de la question numéro 4

	Expert 1	Expert 2	Expert 3
<b>Fréquence</b>	Toujours	Toujours	Toujours
<b>Sévérité</b>	Modéré	Critique	
<b>Pourcentage</b>	100	100	100
<b>Catégorie</b>	Retour d'information	Retour d'information	
<b>Description</b>	« l'utilisateur a enregistré mais il ne se passe rien, pas de feedback »	« en cliquant sur le bouton enregistrer il n'y a aucun réaction du système »	« pas de retour du système sur l'action réalisée »
<b>Justification</b>	« il ne se passe rien après enregistrement »	« l e bouton ne réagit pas et il n'y a pas de réaction du système »	
<b>Recommandation</b>	« il faut un retour d'information qu'on sache que l'enregistrement est terminé »	« il faut introduire un changement d'aspect du bouton et surtout placer un sablier qui indique la réaction du système »	« ajouter feedback »

Parmi les problèmes trouvés par les experts, nous avons pu relever aussi des problèmes de cohérence. Les experts 2 et 3 en ont trouvé un au niveau des saisies d'information. Par exemple, l'ajout d'un domaine consiste à saisir le nom et la description du domaine, puis on clique sur le bouton « ajouter ce domaine » pour l'ajouter dans la liste, tandis que l'ajout d'une fonctionnalité consiste à ajouter une nouvelle fonctionnalité dans la liste, puis on modifie cette nouvelle fonctionnalité en la sélectionnant puis en la modifiant.

Tableau 5.8 : Nombre de problèmes par catégorie et par expert

	Expert 1	Expert 2	Expert 3	Total
<b>Utilisateur</b>	0	0	0	0
<b>Affichage</b>	6	5	2	13
<b>Retour d'information</b>	1	6	1	8
<b>Séquence d'action</b>	1	1	0	2
<b>Texte/Icône</b>	10	2	0	12
<b>Autres</b>	4	0	0	4

### 5.5.3 Discussion sur les deux évaluations

L'évaluation effectuée permet de tirer des conclusions sur un premier ensemble d'améliorations qu'il serait possible d'apporter à CWE.

Ainsi, on a vu que la tâche numéro deux CT-02 (Décrire le système à évaluer) n'a reçu aucun avis « conforme » qui correspond à un déroulement normal de l'exécution ; elle a obtenu un avis « non conforme » et sept avis « sous réserve » qui correspond à une exécution avec erreur, l'erreur ayant pu être contournée. D'après les résultats des évaluations des experts, les erreurs rencontrées et qui peuvent être contournées proviennent peut-être des problèmes d'utilisabilité tels que les problèmes liés à des messages non compris, les problèmes d'organisation d'écran, ou les problèmes d'incohérence des saisies d'information, que nous avons décrits précédemment.

En ce qui concerne les messages non compris, il faut signaler que tous les intervenants impliqués dans cette évaluation ont manipulé l'environnement CWE pour la première fois le jour de celle-ci. En plus, les fonctionnalités relatives à la prise en main de l'évaluation (§3.4.6) n'ont pas encore été implantées, et aucun document sur CWE n'était à la disposition des intervenants lors de cette évaluation ; ce qui peut poser des problèmes de guidage dus à l'interprétation des messages. C'est ce qui est arrivé à l'expert 1 durant l'action 9, à la question 1 : il pense qu'il y a un problème dû à la non compréhension du mot « rôle ». Il a proposé comme recommandation la mise à disposition d'explications sur la notion de « rôle ». Donc si les fonctionnalités relatives à la prise en main de l'évaluation avaient été implantées, l'expert 1 n'aurait pas été amené à signaler ce problème. En effet, selon notre modèle, la prise en main de l'évaluation permet à chaque évaluateur de s'approprier le système à évaluer ainsi que le jargon du domaine de ce système.

Si les problèmes des messages non compris peuvent être résolus par l'implantation des fonctionnalités relatives à la prise en main de l'évaluation, les problèmes de cohérence et les problèmes d'organisation d'écran sont des problèmes persistants qu'il faut prendre en compte.

## 5.6 Conclusion

Nous avons décrit dans ce chapitre l'évaluation mise en œuvre dans le cadre du processus de spécification de l'environnement CWE. La description des stratégies d'évaluation a permis de définir les informations à évaluer, les méthodes d'évaluation à utiliser, de même que les ressources nécessaires à l'évaluation.

Nous avons décidé d'évaluer les programmes et les données puisque les autres types d'informations ne sont pas encore disponibles. La capacité fonctionnelle et la facilité d'utilisation ont été retenues comme caractéristiques logicielles à évaluer. L'évaluation des autres caractéristiques telles que la fiabilité, le rendement, la portabilité n'était pas possible parce que l'environnement CWE n'est pas encore stable. Ensuite, nous avons décrit les déroulements des deux types d'évaluation complémentaire, la première avec des évaluateurs novices, la seconde avec des évaluateurs experts. La description et l'analyse des résultats ont mis en évidence en particulier l'existence de gros problèmes de guidage et de cohérence au niveau de l'utilisation de l'environnement.

Après la mise en évidence de points importants relatifs à chacun des cinq chapitres de ce mémoire, nous allons décrire, dans la conclusion générale, un ensemble de perspectives de recherche, incluant la solution pour pallier ces défauts de guidage et de cohérence.

## Conclusion et perspectives

Dans ce mémoire, nous nous sommes intéressés à l'évaluation des systèmes interactifs et plus particulièrement à l'évaluation par la méthode Cognitive Walkthrough.

La méthode Cognitive Walkthrough a un intérêt théorique par le fait qu'il s'agit de simuler le cheminement cognitif d'un sujet interagissant avec un système interactif. Par contre dans la pratique, on constate qu'il y a des contraintes liées à son utilisation. Ce qui fait que malgré les différentes publications la concernant, peu de praticiens l'utilisent pour l'évaluation des systèmes interactifs dans des cas complexes. En fait, nous avons constaté qu'il n'existe pas d'outil d'aide associé à la méthode. Nous avons présenté dans ce mémoire une spécification d'un environnement d'aide à l'inspection des systèmes interactifs, basé sur la méthode Cognitive Walkthrough. A l'issue de cette spécification une maquette a été réalisée et présentée dans ce mémoire.

Dans le premier chapitre de ce mémoire, nous avons passé en revue différents aspects théoriques concernant les systèmes interactifs et leur évaluation. L'artefact est ainsi décrit sous l'angle de la description de son identité et de celle de sa procédure d'utilisation. En ce qui concerne les tâches, cinq modèles ont été décrits, il s'agit de HTA (Hierarchical Task Analysis), GTA (Groupware Task Analysis), la méthode analytique de description de tâche (MAD), CTT (ConcurTaskTree) et MDA (Modèle de description de l'activité). Concernant l'utilisateur, douze théories et modèles ont été décrits. Ensuite, nous avons décrit de manière globale le processus de modélisation de l'utilisateur. On a vu aussi quelques normes et règles ergonomiques concernant les IHM. Elles ont été classées en quatre catégories selon qu'elles concernent la qualité d'utilisation de l'artefact, la qualité de l'artefact, la qualité du processus de développement et la capacité de l'organisation. Pour terminer le chapitre 1, nous avons décrit l'évaluation comme étant un moyen de vérification de l'adéquation du système à l'utilisateur. La vérification est souvent effectuée en utilisant des critères tels que ceux proposés par Nielsen ou ceux proposés par Bastien et Scapin. Ces deux groupes de critères ont aussi été décrits dans ce chapitre.

Dans le deuxième chapitre, nous avons passé en revue différentes versions et variantes de la méthode Cognitive Walkthrough. Ainsi, nous avons décrit les trois versions initiales ainsi que onze variantes de CW. Pour compléter les descriptions individuelles ces versions et variantes, nous avons présenté quatre synthèses complémentaires de ces différentes versions et variantes. La première concerne les aspects conceptuels, méthodologiques et technologiques ; elle synthétise l'évolution de la méthode selon ces trois aspects. La deuxième synthèse concerne les études comparatives concernant les différents versions et variantes de CW impliquant d'autres méthodes. La troisième synthèse concerne les études non comparatives sur les différentes versions et variantes de Cognitive Walkthrough. Et enfin la dernière synthèse propose à partir des trois premières synthèses une ébauche d'aide au choix d'une méthode parmi les différentes versions et variantes.

On a vu à travers ces quatre synthèses qu'une méthode d'évaluation peut être efficace pour un système donné et qu'elle ne l'est pas forcément pour d'autres systèmes. Ce qui amène certains chercheurs à adapter certaines versions à des types et domaines d'application particuliers. Par exemple, Bligard & Osvalder [2007], Edwards et al. [2008], Niculesen [2008] et Jasper [2008] ont adapté certaines versions et évolutions de CW pour évaluer des IHM dans le domaine médical. Antona, Mourouzis et Stephanisis [2007] ont adapté CW2 pour l'évaluation d'IHM à accès universel. Gonzales, Lorés et Granollers [2007] ont adapté CW3 pour l'évaluation de sites web latino-américains. Ruttkay et Akker [2008] ont adapté CW3 pour l'évaluation de l'interaction entre



humain et humain virtuel<sup>5</sup>. Et enfin Wang (2008) a adapté aussi CW3 dans un processus de conception et d'évaluation de menus dynamiques et intelligents.

Dans le troisième chapitre nous avons spécifié l'environnement CWE. Nous avons identifié deux rôles que les évaluateurs doivent jouer lors d'une session d'évaluation. Le premier rôle est celui de superviseur, l'évaluateur qui joue ce rôle est quelqu'un qui a une vision globale et de haut niveau du produit à évaluer. Tandis que le deuxième rôle est celui d'inspecteur ; l'évaluateur qui joue ce rôle se met à la place des utilisateurs finaux et inspecte les tâches désignées par le superviseur selon la méthode choisie par celui-ci.

Dans ce troisième chapitre, nous avons défini l'évaluation comme étant un processus à trois phases : la phase de préparation, celle d'évaluation et celle d'analyse des résultats.

La phase de préparation a été définie comme étant une activité qui consiste à définir les entrées de l'évaluation et les contextes d'utilisation du système. Durant cette phase, le superviseur effectue les activités suivantes : spécification du système à évaluer, modélisation des utilisateurs finaux du système, choix des scénarii à évaluer, analyse des tâches à évaluer et choix de la méthode d'évaluation à utiliser.

La phase d'évaluation a été définie comme étant une activité permettant dans un premier temps de s'approprier le système à évaluer et les outils à utiliser (environnement CWE, méthode d'évaluation à utiliser). Elle permet ensuite de procéder à l'inspection des tâches par la méthode choisie par le superviseur.

La phase d'analyse des résultats consiste à étudier les données obtenues lors de l'inspection en vue de bien identifier les problèmes et de prendre en compte les recommandations fournies par les inspecteurs.

Après avoir défini le processus d'évaluation, nous avons décrit les différentes entités constituant l'évaluation et les relations entre elles. L'ensemble de ces entités constituent le modèle de domaine, une perspective conceptuelle, non logique, des objets du monde de l'environnement CWE.

Ensuite pour compléter notre spécification, nous avons décrit le comportement dynamique de l'environnement CWE. Nous avons ainsi décrit les différentes activités du processus d'évaluation.

Dans le chapitre quatre, nous avons présenté une maquette de l'environnement CWE en décrivant son architecture ainsi que son interface homme-machine. Avant de décrire l'architecture de l'environnement, nous avons d'abord spécifié les exigences logicielles et les objectifs qui ont un impact significatif sur l'architecture de l'environnement CWE. Ensuite, nous avons décrit les décisions architecturales que nous avons prises.

Pour terminer le chapitre quatre, nous avons décrit les pages écrans des fonctionnalités significatives de l'environnement. Puis nous avons réalisé les cas d'utilisation significatifs correspondants pour illustrer le fonctionnement de chaque page écran.

Dans le chapitre cinq nous avons procédé à l'évaluation de la maquette présentée dans le chapitre quatre. Nous avons décidé d'évaluer la capacité fonctionnelle et l'utilisabilité de l'environnement CWE, et ceci à travers deux groupes d'évaluateurs. Ainsi, nous avons fait tester par le premier groupe constitué d'évaluateurs novices la capacité fonctionnelle de l'environnement par

---

<sup>5</sup> Selon Cassell et al. (2000) (cité dans Ruttlay & Akker (2008), page 90), « Les Humains virtuels sont des personnages de synthèse produit par des moyens de calcul, qui sont destinés à ressembler et communiquer comme de vraies personnes ».

la méthode de test fonctionnel ; puis le deuxième groupe (constitué de trois experts) a utilisé la troisième version de la méthode Cognitive Walkthrough pour détecter les problèmes d'utilisabilité concernant l'environnement.

Après avoir décrit les stratégies d'évaluation ainsi que leur déroulement, nous avons présenté les résultats de l'évaluation. A partir de ces résultats nous avons constaté différents problèmes concernant l'environnement : problèmes de guidage, problème d'organisation d'écran et problèmes de cohérence.

Ces différents problèmes nous amènent à envisager quelques perspectives de recherche pour améliorer l'environnement CWE : certaines sont en cours et d'autres sont envisagées à terme.

Une de nos perspectives à court terme est d'améliorer la maquette actuelle en tenant compte des problèmes évoqués lors de son évaluation. Pour faire face aux problèmes d'organisation d'écran, nous envisageons de la revoir à ce sujet. En ce qui concerne les problèmes de cohérence, nous avons décidé de revoir nos objectifs architecturaux et d'y spécifier la méthode la plus adéquate de saisir l'information.

Pour les problèmes de guidage, nous proposons trois types de solution. La première consiste à fournir un retour d'information à chaque fois que c'est nécessaire. La seconde solution est d'implanter correctement la fonctionnalité « prise en main de l'évaluation ». La troisième solution fait partie d'une perspective à moyen terme consiste à paramétrer les interfaces homme-machine selon les connaissances et expériences des évaluateurs. Pour les évaluateurs qui ne connaissent pas l'environnement, on doit leur proposer des interfaces homme-machine avec plus d'explications dans les cas où cela s'avère nécessaire (par exemple dans certains cas signalés dans les résultats des évaluations des trois experts). Généraliser les explications de certains termes ou de ce qu'il faut faire peut être pénalisant pour les évaluateurs qui ont de bonnes connaissances et des expériences d'utilisation de l'environnement CWE.

Dans le cadre des perspectives à moyen terme, nous envisageons d'étudier le moyen de généraliser un problème. En effet, la méthode Cognitive Walkthrough détecte les problèmes au niveau de l'action. Si un même problème concerne deux actions différentes, le formalisme de la méthode oblige les inspecteurs à décrire ce problème à chaque action. Eviter la ré-description de problèmes peut diminuer le temps d'évaluation.

Et enfin, une autre perspective de recherche à moyen terme, que nous envisageons, concerne les recommandations. Les recommandations sont très importantes car c'est la prise en compte des recommandations qui conduit à l'amélioration de l'Interface Homme-Machine. Mais souvent, elles ne sont pas utiles ni utilisables. Par exemple, en étudiant les recommandations des 17 équipes des professionnels concernant un site Web d'un hôtel, Molich et al. [2007] ont constaté que 14 des 84 recommandations abordant six problèmes d'utilisabilité seulement sont utiles et utilisables. Quatorze recommandations ne sont pas utiles et seize ne sont pas utilisables. Les recommandations peuvent être aussi conflictuelles ou différentes, ce qui peut embarrasser l'équipe de développement lors de la prise en compte de ces recommandations. Un travail de recherche pourrait être mené dans ce domaine.

## Bibliographie

- Abowd, G.D., Coutaz, J., Nigay, L. (1992). Structuring the space of interactive system properties, in the proceedings of the IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction, Ellivuori, Finland, Elsevier Publisher, August 1992.
- Annett, J., Cunningham, D. , Mahias-Jones, P. (2000). A method for measuring team skills, *Ergonomics*, Vol. 43, No. 8, 1076-1094
- Annett, J., Duncan, K.D. (1967). Task Analysis and Training Design, *Occupational Psychology*, 41, 211-221.
- Antona, M., Mourouzis, A. & Stephanidis, C. (2007). Towards a Walkthrough Method for Universal Access Evaluation, *Universal Access in Human Computer Interaction*. In *Coping with Diversity*, Lecture Notes in Computer Science, 4554, Springer Berlin, Heidelberg, 325-334.
- Balbo S. (1994). Evaluation ergonomique des interfaces utilisateur: un pas vers l'automatisation, Thèse de doctorat, Université Joseph Fourier, Grenoble 1.
- Baron, M., Lucquiaud, V., Autard, D., Scapin, D.L. (2006). K-MADe : un environnement pour le noyau du modèle de description de l'activité. Proceedings of the 18th French-speaking conference on Human-computer interaction, Montreal, Canada, April.
- Bass, A., Aspinall, J., Walters, G., Stanton, N.A. (1995). A Software Toolkit For Hierarchical Task Analysis. *Applied Ergonomics*, Vol 26, No 2, 147-151.
- Bastien, J.M.C. (2004). L'inspection ergonomique des logiciels interactifs : intérêts et limites. In *Psychologie ergonomique: tendances actuelles*, Collection Le Travail Humain, Hoc J.M. & Darses F (Eds.), Paris.
- Bastien, J.M.C., Scapin, D.L. (1995). Evaluating a User Interface With Ergonomic Criteria. *International Journal of Human-Computer Interaction*, 7(2), 105–121.
- Bastien, J.M.C., Scapin, D.L. (1993). Critères ergonomiques pour l'évaluation d'interfaces utilisateurs. Rapport technique INRIA n° 156, Juin 1993, INRIA : Le Chesnay.
- Bertelsen, O.W. (2004). The activity walkthrough: an expert review method based on activity theory. In *Proc. NordiCHI*, Tampere, Finland.
- Blackmon, M., Polson, P., Kitajima, M., Lewis, C. (2002). Cognitive Walkthrough for the Web, *Proceeding of CHI*, ACM Press, 463-470.
- Bligard, L.O., Osvalder, A.L. (2007). An Analytical Approach for Predicting and Identifying Use Error and Usability Problem. In *HCI and Usability for Medicine and Health Care*, Lecture Notes in Computer Science, 4799, Springer Berlin, Heidelberg, 427–440.
- Bligard, L.O. (2007). Prediction of Medical Device Usability Problems and Use Errors – An Improved Analytical Methodical Approach, Chalmers University of Technology, Göteborg.

## *Bibliographie*

- Boy, G. (2003a). Documenter un artefact, in G. Boy (Ed), *L'ingénierie cognitive – IHM et cognition* Hermes science publications, 285-318.
- Boy, G. (2003b). Introduction à l'ingénierie cognitive. *Ingénierie cognitive: IHM et cognition*. Paris: Hermès science publications, 23-51.
- Brusilovsky, P. (1996). Adaptive hypermedia, an attempt to analyze and generalize, In Brusilovsky, P., Kommers, P., Streitz, N. (eds.) : *Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science, Vol. 1077*, Springer-Verlag, Berlin, 288-304.
- Bullock, M., Gelman, R., Baillargeon, R. (1982). The development of causal reasoning. In W.J. Friedman (Ed.), *The Developmental Psychology of Time*, New York: Academic Press.
- Calvary G., Coutaz J. (1999). Outils de construction: espace de classification ESPRIT, revue critique et enseignements. Proc. of IHM'99, Montpellier, November.
- Card, S.K., Moran, T.P., Newell, A. (1983). *The Psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.
- Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (2000). *Embodied Conversational Agents*. MIT Press, Cambridge.
- Cockton, G., Lavery, D., Woolrych, A. (2003). Inspection-based evaluations. In J.A. Jacko, A. Sears (Eds.), *The human-computer interaction handbook*, Lawrence Erlbaum Associates, 1118-1138.
- Cuomo, D.L., Bowen, C.D. (1992). Stages of user activity model as basis for user-centered interface evaluation, *Proceedings of the Human Factor and Ergonomics Society 41st annual meeting*, 1254-1258.
- Desjours, C. (1995). *Le Facteur Humain*; Paris, Presses Universitaires de France.
- Desurvise, H.W., Kondziela, J.M., Atwood, M.E (1992). What is gained and lost when using evaluation methods other than empirical testing, In A. Monk, D. Diaper & M.D. Harrison (Eds.), *People and Computers, Proceeding of Human-Computer Interaction*, 89-102.
- Dix, H., Finlay, J., Abowd, G., Beale, R. (1993). *Human-Computer Interaction*, Prentice Hall.
- Duchowski, A.T. (2003). *Eye tracking methodology: theory and practice*, Springer.
- Dumas, J. S., Fox, J. E. (2008). Usability Testing : Current Practice And Future Directions. In *The human-computer interaction handbook*, Julie A. Jacko, Andrew Sears (Eds.), 2nd edition, 1129 – 1149.
- Eden, J (2008). *The Distributed Cognitive Walkthrough: The Impact of Differences in Cognitive Theory on Usability Evaluation*. PhD thesis, Drexel University, USA.
- Eden, J. (2007). Distributed cognitive walkthrough (DCW): a walkthrough-style usability evaluation method based on theories of distributed cognition. *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, June 13-15, 2007, Washington, DC, USA.

## *Bibliographie*

- Edwards, P. J., Moloney, K.P., Jacko, J.A., Sainfort, F. (2008). Evaluating usability of a commercial electronic health record: A case study. *International Journal of Human-Computer Studies*, 66, 718–728.
- Endsley M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37 (1), 32-64.
- Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, volume 47, number 6, June 1954, 381-391.
- Gamboa, F., Scapin D. L. (1997) Editing MAD\* task descriptions for specifying user interfaces, at both semantic and presentation levels. In M. D. Harrison & J. C. Torres (Eds.), *Proceedings of Fourth International Workshop on Design, Specification, and Verification of Interactive Systems (DSV-IS '97)*, 193–208, Berlin: Springer-Verlag.
- Gediga, G., Hamborg, K., Düntsch, I. (1999). The IsoMetrics usability inventory: An operationalisation of ISO 9241-10. *Behaviour and Information Technology*, 18, 151-164.
- Glass, R. L., Vessey, I. (1995). Contemporary Application-Domain Taxonomies, *IEEE Software*, vol. 12, no. 4, 63-76.
- González, M.P., Lorés, J., Granollers, A. (2007). Assessing Usability Problems in Latin-American Academic Webpages with Cognitive Walkthroughs and Datamining Techniques, Usability and Internationalization. *HCI and Culture, Lecture Notes in Computer Science*, 4559, Springer Berlin, Heidelberg, 306–316.
- Gould, J.D. (1988). How to Design Usable Systems. In *Handbook of Human-Computer Interaction*, M. Helander, Editor, Elsevier Science Publishers, Amsterdam, 757-789.
- Gould, J.D., Lewis, C. (1985). Designing for Usability: Key Principles and What Designers Think, *Communications of the ACM*, Vol. 28, 300-310.
- Granollers, T., Lorés, J. (2006). Incorporation of users in the Evaluation of Usability by Cognitive Walkthrough. In Navarro-Prieto R., Vidal J.L. (Eds.), *HCI related papers of interacción 2004*, Springer Netherlands, 243-255.
- Gray, W. D., John, B. E., Atwood, M. E. (1993). Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance. *Human-Computer Interaction*, v.8 (3), 237-309.
- Gray, W.D., Salzman, M. C. (1998). Damaged Merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction*, 11, 203-261.
- Grislin, M., Kolski, C. (1996). Evaluation des Interfaces Homme-Machine lors du développement des systèmes interactifs, *Technique et Science Informatiques*, 15 (3), 265-296.
- Gutwin, C., Greenberg, S. (2000). The mechanics of collaboration: developing low cost usability evaluation methods for shared workspaces. In *Proc. 9th IEEE WETICE 2000*, Gaithersburg MD, 98-103.

## *Bibliographie*

- Helander M., Landauer T.K., Prabhu P. (Eds.) (1997). Handbook of Human-Computer Interaction, Elsevier Science B.V.
- Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, I., Mantei, M., Perlman, G., Strong, G., Verplank, W. (1996). ACM SIGCHI Curricula for Human-Computer Interaction. ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group <http://old.sigchi.org/cdg/cdg2.html>
- Hoc, J.M., Amalberti, R. (1995). Diagnostic et prise de décision dans les situations dynamiques. *Psychologie Française*, 39, 2, 177-192.
- Hollan, J., Hutchins, E., Kirsh, D. (2000). Distributed Cognition: Toward a new foundation for Human-Computer Interaction Research. *ACM Transactions on Computer-Human Interaction*, 7(2), 174-196.
- Huart, J., Kolski, C., Sagar, M. (2004). Evaluation of multimedia applications using inspection methods: The Cognitive Walkthrough case, *Interacting with Computers*, 16, 183-215.
- Hutchins, E.L., Hollan, J.D., Norman, D.A. (1985). Direct manipulation Interface. *Human-Computer Interaction*, 1, 311-388.
- IEC (2004). Medical electrical equipment - Part 1-6: General requirements for safety - Collateral standard: Usability IEC.
- Jacko J.A., Sears A. (Eds.) (2002). The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications (human factors and ergonomics), Lawrence Erlbaum Associates.
- Jacobsen, N.E., John, B.E. (2000). Two Case Studies In Using Cognitive Walkthrough For Interface Evaluation, Technical Report Carnegie Mellon University, Pittsburgh, CMU-CS-00-132/CMU-HCI-00-100.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999). The Unified Software Development Process. Reading, MA.: Addison –Wesley.
- Jaspers, M.W.M. (2008). A comparison of usability methods for testing interactive health technologies: Methodological aspects and empirical evidence. *International Journal of Medical Informatics*, 78 (5), 340-353.
- Jeffries, M., Miller, J.R., Wharton, C., Uyeda, K.M. (1991). User interface evaluation in real world: a comparison of four techniques. In *Proceedings of ACM Computer-Human Interaction*, ACM Press, 119-124.
- Jeffries, R. (1994). Usability Problem Reports: Helping Evaluators Communicate Effectively with Developers. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*, New York: John Wiley, 273-294.
- John, B. E. (1995) Why GOMS? *Interactions*, vol. 2, no. 4, 80-89.

## *Bibliographie*

- John, E.B., Marks, S.J. (1997). Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology*, 16(4/5), 188-202.
- John, B.E., Kieras, D.E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer Human Interaction*, 3, 320-351.
- Johnson, P. (1992). *Human-computer interaction: Psychology, task analysis and software engineering*. London: McGraw-Hill.
- Kaptelinin, V. (1996). Computer-Mediated Activity : Functional Organs in Social and Developmental Contexts". In Nardi B. (ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, MIT Press, 45-68.
- Karat, C.M., Cambell, R., Fiegel, T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proc. ACM CHI*, 397-404.
- Kato, T., Hiro, M. (2006). Beyond Perceivability: Critical Requirements for Universal Design of Information. 8th Annual ACM Conf. on Assistive Technologies, ACM Press, 287-288.
- Kato, T., Hori, M. (2005). Articulating the cognitive walkthrough based on an extended model of HCI. *Proc. HCI International*, Las Vegas.
- Kieras, D. (1985). An approach to the formal analysis of user complexity. *International Journal of Human-Computer Studies*, 22, 365-394.
- Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), *Handbook of human-computer interaction*, Amsterdam, The Netherlands: North Holland.
- Kieras, D. E., Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*. 22, 365-394.
- Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-Integration model. *Psychological Review*, 95 (2), 63-182.
- Kirwan, B., Ainsworth, L. K. (1992). *A guide to task analysis*. Taylor and Francis, London.
- Kitajima, M., Polson, P.G. (1997). A comprehension based model of comprehension. *Human-Computer Interaction*, Vol. 12, 345-389.
- Kitajima, M. (2006). Cognitive Walkthrough for the web. *International Encyclopedia of Ergonomics and Human Factors* (2nd Edition), Chapter 216.
- Kitajima, M., Blackmon, M.H., Polson, P. (2000). A Comprehension-based model of Web navigation and its application to Web usability analysis. In *People and Computer XIV*, 357-373.
- Kobsa, A., Wahlster, W. (1990). User models in dialog systems. In: Kobsa, A., Wahlster, W. (eds.): *User Models in Dialog Systems*, Berlin: Springer, 4-34.

## Bibliographie

- Kolski, C., Ezzedine, H., Abed, M. (2001). Développement du logiciel : des cycles classiques aux cycles enrichis sous l'angle des IHM. In Kolski C. (Ed.), *Analyse et Conception de l'IHM. Interaction Homme-machine pour les SI*, Vol. 1, Hermès, Paris, pp. 23-49.
- Kruchten, P. (2000). *The Rational Unified Process – An Introduction*, 2nd edition, Reading, MA: Addison-Wesley.
- Kules, B. (2000). *User Modeling for Adaptive and Adaptable Software Systems*, [http :  
//www.otal.umd.edu/UUGuide/wmk/](http://www.otal.umd.edu/UUGuide/wmk/).
- Landauer, T.K., Dumas, S.T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240.
- Larman, C. (2005). *UML 2 et les Design Patterns*, Troisième édition, Pearson Education.
- Leite, J.C.S.P., Franco, A.P.M. (1993). A strategy for conceptual model acquisition. In *IEEE international symposium on requirements engineering*, IEEE Computer Society Press, 243-246.
- Leite, J.C.S.P., Hadad, G.D.S., Doorm, J.H., Kaplan, G.N. (2000). A Scenario Construction Process. *Requirements Engineering*, 5, 38-61.
- Lewis, C., Wharton, C. (1997). Cognitive Walkthrough. In *Handbook of Human-Computer Interaction*, M. Helander, T.K. Landauer, P. Prabhu (eds.), Elsevier Science B.V, 717-732
- Lewis, C. (1986). A model of Mental Model Construction. In *Proceedings of ACM Computer-Human Interaction*, 306-313.
- Lewis, C. (1988). Why and How to learn why: analysis-based generalization of procedure. *Cognitive Science*, 12, 211-225.
- Lewis, C., Polson, P., Wharton, C., Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up and-use interface. In *Proceeding of ACM Computer-Human Interaction*, ACM Press, 235-242.
- Lonchamp, J. (2003). *Les travaux coopératif et ses technologies*, Hermes science Lavoisier.
- Lucquiaud, V. (2005). Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateurs. *Proceedings of the 17th international conference Francophone sur l'Interaction Homme-Machine*, 83-90, September 27-30, Toulouse, France
- Mack, R.L., Nielsen, J. (1994). Executive Summary. In Nielsen J., R.L. Mack (Eds.), *Usability Inspection Methods*, Elsevier, 4-23.
- Mahatody, T., Kolski, C., Sagar, M. (2009). CWE : Assistance environment for evaluation operating a set of variations of cognitive walkthrough ergonomic inspection method. In D. Harris (Ed.), *Engineering Psychology and Cognitive Ergonomics*, HCII 2009, LNAI 5639, 52–61.



## Bibliographie

- Mahatody, T., Sagar, M., Kolski, C. (2007a). Cognitive Walkthrough pour l'évaluation des IHM : synthèse des extensions et évolutions conceptuelles, méthodologiques et technologiques. Proceedings of IHM 2007, 19ème Conférence de l'Association Francophone d'Interaction Homme-Machine (Paris, France, 13-15 novembre 2007), International Conference Proceedings Series, ACM Press, Paris, 143-150.
- Mahatody, T., Sagar, M., Kolski, C. (2007b). Cognitive Walkthrough for HCI evaluation: basic concepts, evolutions and variants, research issues. Proceedings EAM'07 European Annual Conference on Human-Decision Making and Manual Control, Technical University Of Denmark, Lyngby, juin.
- Mahatody, T., Sagar, M., Kolski, C. (2010). State of the art on the Cognitive Walkthrough method, its variants and evolutions. International Journal of Human Computer Interaction, sous presse.
- Mariage, C., Vanderdonckt, J., Pribeanu, C. (2005). State of the Art of Web Usability Guidelines. In The Handbook of Human Factors in Web Design, Lawrence Erlbaum Associates, Mahwah, 688–700.
- Marshall, C., Nelson, C., Gardiner, M. M. (1987). Design guidelines. In M. M. Gardiner & B. Christie (Eds.), Applying cognitive psychology to user-interface design, Chichester, UK: Wiley, 221-278.
- MoD (2000). Human Factor Integration: An introductory guide. London, HMSO.
- Molich, R., Jeffries, R., Dumas, J. (2007). Making Usability Recommendations Useful and Usable Journal of Usability Studies, 2(4), August, 162-179.
- Morand, B. (2001). Cinq approches pour définir le concept d'architecture logicielle. Journée du PRC GdR I3, Lyon.
- Muller, P.A., Gaertner, N. (2002). Modélisation objet avec UML, Eyrolles.
- Nardi, B.A. (1996). Context and Consciousness: Activity Theory and Human-Computer Interaction. M.I.T. Press, Cambridge.
- Niculescu, A. (2008). Affordances in Conversational Interactions with Multimodal QA Systems. In HCI and Usability for Education and Work, Lecture Notes in Computer Science, 5298, Springer Berlin, Heidelberg, 221-236.
- Nielsen, J., Mack, R. (1994). Usability Inspection Methods. John Wiley & Sons, New York.
- Nielsen, J., Molich, R. (1990). Heuristic Evaluation of User Interfaces, Proceedings of CHI conference on Human Factors in Computing System, 249-256.
- Nielsen, J. (1992). Finding Usability Problems Through Heuristic Evaluation. In Proc. ACM CHI, 373-380.

## *Bibliographie*

- Norman, D.A. (1986). Cognitive engineering. In D.A. Norman & S.W. Draper (Eds.), *User centered systems design: New perspectives in human-computer interaction*, Lawrence Erlbaum Associates, Hillsdale, 31-61.
- Norman, D.A. (1988). *The psychology of everyday things*. Basic Books.
- Norman, D.A. (1999). Affordances conventions and design. *Interactions*, 6 (3), 38-42.
- Oppermann, R., Reiterer, R. (1997). Software Evaluation Using the 9241 Evaluator Usability Evaluation Methods. *Behaviour and Information Technology*, 16 (4/5) 232-245.
- Paternò F. (2000). *Model-Based Design and evaluation of Interactive Applications*. Springer-Verlag Eds.
- Pierret-Goldreich, C., Delouis, I., Scapin, D.L. (1989). *Un Outil d'Acquisition et de Représentation des Taches Orienté-Objet*, rapport de recherche INRIA no 1063, Programme 8 : Communication Homme-Machine, août.
- Pierroti, D. (2004). *Heuristic Evaluation - A System Checklist*, <http://www.stcsig.org/usability/topics/articles/he-checklist.html>
- Pinelle, D., Gutwin, C. (2002). Groupware walkthrough: Adding context to groupware usability. In *Proceeding of ACM CHI*, ACM Press, 455-462.
- Pinelle, D., Gutwin, C., Greenberg, S. (2003). Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. *Transactions on Computer-Human Interaction (TOCHI)*, 10(4), 281-311
- Pohl, W. (1997). *Logic-based representation and reasoning for user modeling shell systems*. Ph.D. thesis, University of Essen.
- Polson, P., Lewis, C., Rieman, J., Wharton, C. (1992). Cognitive Walkthrough: a method for theory-based evaluation of user interface. *International Journal of Man-Machine Studies*, 36, 741-773.
- Polson, P.G., Lewis, C.H. (1990). Theory-based design for easily learned interfaces. *Human-Computer Interaction*, 5 (5), 191-220.
- Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. Elsevier Science Inc. New York, NY, USA
- Razmerita, L. (2003). *Modèle Utilisateur et Modélisation Utilisateur dans les Systèmes de Gestion des connaissances : une approche fondée sur les ontologies*. Thèse de doctorat, Université de Toulouse III.
- Reiterer, H., Oppermann, R. (1995). Standards and software-ergonomic evaluation. In Y. Anzai, K. Ogawa, and H. Mori (eds.), *Symbiosis of Human and Artefact: Human and Social Aspects of Human-Computer Interaction*, Elsevier Science B.V., Amsterdam, 361 - 366.
- Rich, E. (1979). User Modeling via stereotypes. *Cognitive science*, 3, 329-354.

## *Bibliographie*

- Richard, J.F. (1990). *Les activités mentales*, PUF, Paris
- Rieman, J., Davies, S., Hair, D.C., Esemplare, M., Polson, P., Lewis, C. (1991). An Automated Walkthrough: Description and Evaluation. In *Proceeding of ACM Computer-Human Interaction*, 427-428.
- Riihihaho, S. (2000). *Experiences with usability evaluation methods*. Licentiate's thesis, Helsinki Univ. Of Technology, Laboratory of Information Processing Science.
- Ritter, F. E., Young, R. M. (2001). Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design. *International Journal of Human Computer Interaction Studies*, 55, 1-14
- Rizzo, A., Mandrigiani, E., Andreadis, A. (1997). The AVANTI Project: Prototyping and Evaluation with a Cognitive Walkthrough Based on the Norman's Model of Action. In *Proceedings of Designing Interactive Systems: Processes, Practices, Methods, & Techniques*, 305-309.
- Robert, J.M. (2003). Que faut-il savoir sur les utilisateurs pour réaliser des interfaces de qualité? In Boy G. (Ed.), *Ingénierie cognitive: IHM et cognition*, Paris: Hermès science publications, 249-283.
- Rowley, D.E. Rhodes, D.G. (1992). The Cognitive Jogthrough : A fast-paced user interface evaluation procedure. In *proceedings of ACM Computer-Human Interaction*, 398-395.
- Ruttkay, Z., Akker, R. (2008). Affordances and Cognitive Walkthrough for Analyzing Human-Virtual Human Interaction. In *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction*, Lecture Notes in Computer Science, 5042, Springer, Berlin, Heidelberg, 90-106.
- Ryu, H., Monk, A. (2004). Analysing interaction problems with cyclic interaction theory: Low-level Interaction Walkthrough. *Psychology J.*, 2 (3), 304-330.
- Ryu, H. (2008). Collective Web Usability analysis: cognitive and activity walkthroughs. *International Journal of Web Engineering and Technology*, 4(3), 286-312.
- Scapin, D. L., Bastien, J. M. C. (2001). Analyse des tâches et aide ergonomique à la conception : l'approche MAD\*. In C. Kolski (Ed.), *Analyse et conception de l'IHM : Interaction homme-machine pour les systèmes d'information*, Vol. 1, 85–116, Paris: Edition Hermès.
- Scapin, D.L., Leulier, C., Vanderdonckt, J., Mariage, C., Bastien, Ch., Farenc, Ch., Palanque, Ph., and Bastide, R. (2000). A Framework for Organizing Web Usability Guidelines. In Ph. Kortum and E. Kudzinger (eds.), *Proc. of 6th Conf. on Human Factors and the Web HFWeb'2000* (Austin, 19 June 2000), University of Texas, Austin, 2000, accessible à <http://www.tri.sbc.com/hfweb/scapin/Scapin.html>
- Sears, A., Hess, D.J. (1999). Cognitive Walkthroughs: Understanding the Effect of Task-Description Detail on Evaluator Performance. *International Journal of Human-Computer Interaction*, 11 (3), 185-200.

## *Bibliographie*

- Sears, A. (1997). Heuristic Walkthroughs: Finding the problems without the noise. *International Journal of Human-Computer Interaction*, 9, 213-234.
- Sears, A. (2003). Testing and evaluation. In J.A. Jacko, A. Sears (Eds.), *The human-computer interaction handbook*, Lawrence Erlbaum Associates, 1091-1092.
- Senach, B. (1990). Evaluation ergonomique des interfaces homme-machine: une revue de la littérature. Rapport n°1180, INRIA.
- Shackel B. (1991). Usability, context, Framework, Definition, Design and Evaluation. In Shackel B. and Richardson S. (Eds.), *Human Factors for Informatics Usability*, Cambridge: Cambridge University Press, 21-38.
- Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction: Third Edition*. Addison-Wesley Publ. Co., Reading, MA.
- Scholtz, J., & Morse, E. (2002). A new usability standard and what it means to you. *SIGCHI Bulletin*, 10-11.
- Shultz, T.R., Ravinsky, F.B. (1977). Similarity as a principle of causal inference. *Child Development*, 48, 1552-1558.
- Spencer, R (2000). The Streamlined Cognitive Walkthrough method, working around social constraints encountered in a software development company. In *Proceedings of ACM CHI*, ACM Press, 353-359.
- Stanton, N. A. (2006). Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics*, 37, 55-79.
- Stephanidis, C., Akoumianakis, D. (1999). Accessibility Guidelines and Scope of Formative HCI Design Input: Contrasting Two Perspectives. *User Interfaces for All, Proceedings of the 5th ERCIM Workshop, Dagstuhl, Germany*, 119-130.
- Sweeney, M., Maguire, M., Shackel, B. (1993). Evaluating user-computer interaction: a framework. *International Journal of Man-Machine Studies*, 38, 689-711.
- Van der Veer G.C., Van Welie M. (1999). Groupware Task Analysis. In: *Tutorial Notes for the CHI99 workshop Task Analysis Meets Prototyping: Towards seamless UI Development*, Pittsburgh PA, USA.
- Van der Veer, G.C., van Velie, M. (2000). M. Task-Based Groupware design: Putting Theory into Practice. In *Proc. DIS'2000*, 326-337.
- Van der Veer, G.C., G.C., Lenting, B.F., Bergevoet, B.A.J (1996). GTA:Groupware Task Analysis – Modeling Complexity. *Acta Psychologica*, 91, 297-322.
- vanWelie, M., van der Veer, G. C. (1998). EUTERPE: Tools support for analyzing cooperative environments. In T. R. G. Green, L. Bannon, C. P. Warren, & J. Buckkleys (Eds.), *Cognition and co-operation*, 25–30, Roquencourt INRIA.

## *Bibliographie*

- Virzi, R.A. (1997). Usability inspection methods. In M. Helander, T.K. Landauer, P. Prablhu (Eds.). *Handbook of Human-Computer Interaction*, Elsevier, Amsterdam, 705-715.
- Vredenburg, K., Mao, J., Smith, P. et Carey, T. (2002). A Survey of User-Centered Practice. *Human Factors in Computing Systems, CHI'2002 Conference Proceedings*, ACM, New York, 471-478.
- Wang, X. (2008). Design and Evaluation of Intelligent Menu Interface through Cognitive Walkthrough Procedure and Automated Logging for Management Information System. In *Computer Supported Cooperative Work in Design IV, Lecture Notes in Computer Science*, 5236, Springer Berlin, Heidelberg, 408-418.
- Wharton, C., Bradford, J., Jeffrey, R., Franzke, M. (1992). Applying cognitive walkthrough to more complex user interfaces: Experiences, issues, and recommendations. *Proceedings of Computer-Human Interaction*, ACM Press, 381-388.
- Wharton, C., Rieman, J., Lewis, C., Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods*, John Wiley & Sons, New York, 105-140.
- Zhang, J., Norman, D.A. (1994). Representations in Distributed Cognitive Tasks. In *Cognitive Science*, 18, 87-122.

# Annexes

## Annexe A Plan d'évaluation

### 1-Introduction

#### 1.1-Objectif du document

Ce document présente l'organisation, l'environnement et la planification des activités d'évaluation de l'environnement CWE. Les fonctionnalités à tester ainsi que les tâches correspondantes sont décrits dans ce document.

#### 1.2-Portée du document

Ce document concerne les fonctionnalités significatives de l'environnement CWE. Ces fonctionnalités couvrent les trois phases de l'évaluation: préparation, inspection et analyse des résultats.

#### 1.3-Documents de référence

Cahier d'évaluation

### 2-Exigences de test

Cette section dresse la liste des exigences retenues pour le test. Dans le cadre de ce présent plan de test, les exigences de test concernent donc les fonctionnalités significatives ainsi que la facilité d'utilisation de l'environnement CWE.

EX-F-01	identification de l'évaluateur
EX-F-02	spécification du système à évaluer
EX-F-03	modélisation de l'utilisateur
EX-F-04	choix de scénarii à évaluer
EX-F-05	analyse des tâches
EX-F-06	choix de méthode d'évaluation
EX-F-07	inspection des tâches
EX-F-08	analyse des résultats
EX-U-01	Facilité d'utilisation

## 3-Description du logiciel

### 3.1-Identité du logiciel

**Nom du logiciel :** Le logiciel s'appelle Cognitive Walkthrough Environment (CWE)

**Domaine d'application :** C'est un logiciel pour l'évaluation des IHM.

**Objectif du logiciel :** CWE est un environnement comportant plusieurs outils et aides pour l'évaluation des IHM basée sur la méthode Cognitive Walkthrough.

### 3.2-Information

**Programme :** cwe.jar

**Données :** Base de données mysql cwenv.

## 4-Stratégie générale de test

Cette section présente la stratégie de test mise en place pour la validation de l'environnement CWE. Il précise notamment, pour chaque caractéristique logicielle [selon la norme ISO 9126] à tester, les informations logicielles concernées ainsi que les techniques de test retenues.

### 4.1-Information à tester

Les informations à tester sont les programmes et les données. Les autres informations telles que la documentation utilisateur, la documentation technique ainsi que les codes sources pourront faire l'objet d'une autre session d'évaluation. De même, les caractéristiques logicielles à étudier sont la capacité fonctionnelle et la facilité d'utilisation. Les autres caractéristiques telles que la fiabilité, le rendement, la maintenabilité et la portabilité pourront faire l'objet d'une autre évaluation ultérieure.

Information	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Programmes et données	X		X			
Document utilisateur						
Document technique						
Code source						

### 4.2-Techniques de test

Pour la caractéristique logiciel capacité fonctionnelle, on va utiliser la méthode de test fonctionnel. Cette méthode permet de vérifier les utilisations normales des exigences. Tandis que pour la facilité d'utilisation, on va utiliser la méthode Cognitive Walkthrough

## 5-Spécification de test

Cette section raffine la stratégie de test précédente. Elle précise pour chaque exigence les cas de test à exécuter. Ainsi, le tableau suivant décrit les cas de test pour chaque exigence de test:

EX-F-01	CT-01
EX-F-02	CT-02
EX-F-03	CT-03
EX-F-04	CT-04
EX-F-05	CT-05
EX-F-06	CT-06
EX-F-07	CT-07
EX-F-08	CT-08
EX-U-01	INSPECTION

## 6-Organisation de test

Cette section décrit l'organisation mise en place pour réaliser ce plan de test. Elle précise entre autre les fonctions et les ressources allouées ainsi que les tâches à exécuter et les responsables associés.

Fonctions et ressources allouées

Le tableau ci-dessous indique les différentes fonctions requises pour exécuter le présent plan de test ainsi que les ressources initialement allouées pour chacune de ces fonctions.

Fonctions	Ressources allouées	Organisation
Représentant Utilisateurs	Christophe KOLSKI, Mouldi SAGAR	UVHC
Chef de projet	Thomas MAHATODY	UVHC
Exécutant	Thomas MAHATODY, Christophe KOLSKI, Mouldi SAGAR, 16 étudiants en master informatique	UVHC

## 7-Environnement de test

Cette section décrit l'environnement requis pour la réalisation de ce plan.

Configuration matérielle

Chaque groupe d'exécutant doit posséder au moins un PC. L'idéal est d'en posséder deux : sur l'un s'exécutera l'environnement et sur l'autre se déroule l'IHM à évaluer.

Configuration logicielle



La configuration logicielle suivante est requise pour exécuter le présent plan de test :

- Java 2 Runtime Environement SE de version supérieur ou égale à v1.4.2
- SGBD mysql de version supérieur ou égale à 5.1.30
- Windows XP

## Annexe B Cahier d'évaluation Test fonctionnel

### 1-Introduction

#### 1.1-Objectif du document

Ce document présente les cas de test de l'environnement CWE conformément aux spécifications de test décrit dans le Plan d'évaluation. Elle doit réaliser les scénarii décrits dans la section 3 ci-dessous, sur un scénario issu du site de la SNCF.

#### 1.2-Références

Documents de référence  
Plan d'évaluation

### 2-Équipe de test

Fonction	Nom et signature
Chef de projet	Thomas MAHATODY
Exécutant	
Exécutant	

### 3-Scenarii

Cas de test	CT-FONC-01		
Titre	Spécifier le profil de l'évaluateur		
Objectif	Vérifier si le système permet de spécifier le profil de l'évaluateur		
Exigence de test	Spécifier le profil de l'évaluateur		
Rôle	Superviseur ou Inspecteur		
Exécutable	cwe.jar		
Procédure de test			
Précondition	Système lancé		
N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Fichier »	Un menu contenant « Profil utilisateur » se déroule	
2	Cliquer sur « Profil utilisateur »	L'écran « Écran d'identification utilisateur » s'affiche	
3	Cliquer sur l'onglet « Identification »	Le formulaire « Information d'identification » s'affiche	
4	Remplir le formulaire		
5	Cliquer sur le bouton « Enregistrer »	Un message « Login info enregistrés » s'affiche	
6	Cliquer sur le bouton « OK »	Le message disparaît	
7	Cliquer sur l'onglet « Connaissance Informatique »	Le formulaire « Expérience en informatique » s'affiche	
8	Remplir le formulaire		
9	Cliquer sur le bouton « Enregistrer »	Un message « Connaissances informatiques enregistrés » s'affiche	
10	Cliquer sur le bouton « OK »	Le message disparaît	
11	Cliquer sur l'onglet « Expérience en CW »	Le formulaire « Expérience en CW » s'affiche	
12	Remplir le formulaire		
13	Cliquer sur le bouton « Enregistrer »	Un message « Expérience en CW enregistrés » s'affiche	
14	Cliquer sur le bouton « OK »	Le message disparaît	
15	Cliquer sur l'onglet « Expérience en CWE »	Le formulaire « Expérience en CWE » s'affiche	
16	Remplir le formulaire		
17	Cliquer sur le bouton « Enregistrer »	Un message « Expérience en CWE enregistrés » s'affiche	
18	Cliquer sur le bouton « OK »	Le message disparaît	

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			

Annexe

Cas de test	CT-FONC-02
Titre	Décrire le système à évaluer
Objectif	Vérifier si le système permet de décrire le système à évaluer
Exigence de test	Décrire le système à évaluer
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Spécification » se déroule	
2	Cliquer sur « Spécification »	L'écran « Écran de description de système » s'affiche	
3	Cliquer sur l'onglet « Système »	Le formulaire de description du système s'affiche	
4	Cliquer sur « Créer »	L'onglet « Identification » est initialisé	
5	Saisir le nom et la description du système		
6	Choisir le domaine dans la liste ou Saisir le nom de domaine et sa description puis cliquer sur le bouton « Ajouter ce domaine »	Le nouveau domaine apparaît dans la liste des domaine si on a cliqué sur le bouton « Ajouter ce domaine »	
7	Choisir la motivation du système <sup>6</sup>		
8	Choisir le contexte d'utilisation du système <sup>7</sup>		
9	Saisir un rôle		
10	Cliquer sur le bouton « Ajouter un rôle »	Le nouveau rôle apparaît dans la liste des rôles	
11	Cliquer sur le bouton « Ajouter une fonctionnalité »	La fonctionnalité « Nouvelle fonctionnalité » apparaît dans la liste des fonctionnalités	
12	Sélectionner la « Nouvelle fonctionnalité »	La fonctionnalité sélectionnée apparaît dans le formulaire à droite de la liste	
13	Modifier la nouvelle fonctionnalité		
14	Cliquer sur le bouton « OK »	La nouvelle fonctionnalité est modifié dans la liste des fonctionnalités	
15	Répéter 13 et 14 jusqu'à ce qu'on n'a plus de fonctionnalité à ajouter		
16	Cliquer sur le bouton « Enregistrer »	Un message « Expérience en CWE enregistré » s'affiche	
17	Cliquer sur le bouton « OK »	Le message disparaît	

<sup>6</sup>Deux classes de motivation : la motivation industrielle et commerciale et la motivation domestique.

<sup>7</sup>Les 3 classes de contexte sont : le contexte embarqué, le contexte portable et les autres qui ne sont pas ni embarqué ni portable. Le contexte embarqué concerne les interfaces intégrées ou incorporées avec le dispositif. C'est le cas des appareils domestiques comme les fours, les machines à laver. Quant au contexte portable, il concerne les interfaces accessibles via des dispositifs portables.

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			

Annexe

Cas de test	CT-FONC-03
Titre	Décrire les connaissances des utilisateurs concernant le système
Objectif	Vérifier si le système permet de décrire les connaissances des utilisateurs le système à évaluer
Exigence de test	Décrire les connaissances des utilisateurs concernant le système
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Modélisation utilisateur se déroule	
2	Cliquer sur « Modélisation utilisateur »	L'écran « Écran d'identification utilisateur » s'affiche	
3	Cliquer sur l'onglet « Identité »	Le formulaire « Identité de l'utilisateur » s'affiche	
4	Saisir le nom du système		
5	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe , la description est affichée et le formulaire identité de l'utilisateur est initialisé	
6	Cliquer sur le bouton « ajouter utilisateur »	Un utilisateur « Nouveau utilisateur » s'affiche dans la liste des utilisateurs	
7	Sélectionner « Nouveau utilisateur » dans la liste	« Nouveau utilisateur » s'affiche dans le formulaire identité de l'utilisateur	
8	Modifier le nouveau utilisateur dans le formulaire		
9	Cliquer sur le bouton « Enregistrer »	Le nouveau utilisateur est mis à jour dans la liste	
10	Répéter 6-9 jusqu'à ce qu'on n'a plus d'utilisateur à ajouter		

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			



Annexe

Cas de test	CT-FONC-04
Titre	Choisir les scenarii à évaluer
Objectif	Vérifier si le système permet de décrire les scenarii à évaluer
Exigence de test	Choisir les scenarii à évaluer
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Choix de scenarii » se déroule	
	Cliquer sur « Choix de scenarii »	L'écran « Écran de description de scénario » s'affiche	
2	Cliquer sur l'onglet « Description scénario »	Le formulaire « Description de scénario» s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe , la description est affichée, les fonctionnalités du système s'affichent dans l'entête de la liste des scenarii	
5	Cliquer sur le bouton « ajouter scénario »	Un scénario « Nouveau scénario » s'affiche dans la liste des scenarii	
6	Sélectionner le nouveau scénario	Le nouveau scénario apparaît dans le formulaire en bas de la liste de scénario	
7	Modifier le nouveau scénario		
8	Choisir les acteurs du scénario en sélectionnant un acteur puis en cliquant sur le bouton « > »	L'acteur sélectionné passe de la liste des acteurs vers la liste des acteurs choisis	
9	Cliquer sur le bouton « Valider »	Le nouveau scénario est mis à jour dans la liste	
10	Dans les colonnes fonctionnalités Saisir « Oui » si le scénario concerne la fonctionnalité.		
11	Répéter 5-10 jusqu'à ce que l'ensemble de scénario couvre le maximum de fonctionnalités		
12	Cliquer sur le bouton « Enregistrer les scenarii »	Un message « Scenarii enregistrés » s'affiche	
12	Cliquer sur le bouton « OK »	Le message disparaît	

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			

Cas de test	CT-FONC-05
Titre	Analyser les tâches
Objectif	Vérifier si le système permet de décomposer les scenarii en suite d'action
Exigence de test	Analyser les tâches
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Analyse de tâches » se déroule	
2	Cliquer sur « Analyse de tâches »	L'écran « Écran d'analyse de tâche » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe , les scenarii apparaissent dans la liste déroulante	
5	Choisir un scénario dans la liste déroulante		
6	Cliquer sur un scénario ou tâche ou sous tâche dans la liste des tâches	L'objet sélectionné apparaît dans le formulaire de droite	
7	Remplir le formulaire de droite		
8	Cliquer sur valider	L'objet modifié est mis à jour dans la liste des tâches	
9	Décomposer l'objet sélectionner en ajoutant d'objet fils (ajouter une tâche fille, sous-tâche fille, sous , etc.)	Un nouveau objet fils apparaît dans la liste hiérarchisée.	
10	Répéter 6-9 jusqu'à ce que tout est décomposé en action		
11	Cliquer sur le bouton « Enregistrer »	Un message « Scénario enregistrés » s'affiche	
12	Cliquer sur le bouton « OK »	Le message disparaît	
13	Reprendre à partir de 5 jusqu'à ce tous les scenarii sont décomposés		

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			

Cas de test	CT-FONC-06
Titre	Choisir les méthodes d'évaluation
Objectif	Vérifier si le système permet de choisir une méthode d'évaluation et de l'affecter à une tâche et un évaluateur
Exigence de test	Choisir les méthodes d'évaluation
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Analyse de tâches » se déroule	
2	Cliquer sur « Analyse de tâches »	L'écran « Écran d'analyse de tâche » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, les scénarii apparaissent dans la liste déroulante	
5	Choisir un scénario dans la liste déroulante	La description du scénario apparaît. Les tâches et les sous-tâches apparaissent dans la liste des tâches	
6	Sélectionner une sous-tâche		
7	Cliquer sur la partie évaluateur pour afficher la liste déroulante des évaluateurs	La liste déroulante des évaluateurs apparaît	
8	Choisir un évaluateur		
9	Cliquer sur la partie méthode pour afficher la liste déroulante des méthodes	La liste déroulante des méthodes apparaît	
10	Choisir une méthode		
11	Cliquer sur le bouton « Ajouter une session » si on veut ajouter une autre session d'évaluation	Une nouvelle ligne correspondant à la sous-tâche sélectionnée apparaît	
12	Répéter 6-11 jusqu'à ce que qu'on a le nombre de session souhaité pour chaque sous-tâche		
13	Cliquer sur le bouton « Enregistrer »	Un message « Scénario enregistré » s'affiche	
14	Cliquer sur le bouton « OK »	Le message disparaît	

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			

Annexe

Cas de test	CT-FONC-07
Titre	Inspecter les tâches
Objectif	Vérifier si le système permet d'inspecter les tâches
Exigence de test	Inspecter les tâches
Rôle	Inspecteur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Inspection »	Un menu contenant « Évaluer » se déroule	
2	Cliquer sur « Évaluer »	L'écran « Écran d'inspection » s'affiche et le nom de l'évaluateur connecté ainsi que les tâches qu'il doit évaluer s'affichent aussi	
5	Choisir une tâche dans la liste déroulante	La première action et la première action apparaissent	
6	Choisir la réponse et saisir la justification		
7	Si la réponse est non, remplir le formulaire de problème d'utilisabilité		
8	Cliquer sur le bouton « Ajouter » si on veut ajouter un autre problème	Le formulaire de problème est initialisé	
9	Remplir le formulaire de problème d'utilisabilité		
10	Répéter 8-9 jusqu'à ce qu'on n'a plus de problème		
11	Cliquer sur le bouton « Question suivante »		
12	S'il y a encore de question, reprendre à partir de 6		
	S'il n'y a plus de question, cliquer sur le bouton « Action suivante »		
	Reprendre à partir de 6		
	Cliquer sur le bouton « Terminer »		
	S'il y a encore de tâche, reprendre à 5		

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			



Annexe

Cas de test	CT-FONC-08
Titre	Calculer les tableaux de bord
Objectif	Vérifier si le système permet de calculer les tableaux de bord après l'inspection
Exigence de test	Calculer les tableaux de bord
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Analyse »	Un menu contenant « Calculer tableau de bord » se déroule	
2	Cliquer sur « Calculer tableau de bord »	L'écran « Écran tableau de bord résultat » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, les scénarii apparaissent dans la liste déroulante	
5	Choisir un scénario dans la liste déroulante	La description du scénario apparaît. Les sous-tâches évaluées ainsi que les méthodes et les évaluateurs apparaissent chacune dans une liste déroulante	
8	Choisir une sous tâches, une méthode et un évaluateur	La liste des problèmes apparaît	
9	Cliquer sur l'onglet « Synthèse des problèmes par catégorie » pour voir la synthèse de problème par catégorie avec le graphique	La synthèse des problèmes par catégorie apparait	
10	Cliquer sur l'onglet « Synthèse des problèmes par évaluateur » pour voir la synthèse de problème par évaluateur avec le graphique	La synthèse des problèmes par évaluateur apparait	

Annexe

Rapport de test			
Testé par :		Date:	
Critère de succès			
<input type="checkbox"/>	Conforme : l'exécution s'est déroulée normalement		
<input type="checkbox"/>	Sous réserve : l'exécution s'est déroulée jusqu'à la fin avec des erreurs qui ont pu être contournées		
<input type="checkbox"/>	Non conforme : l'exécution ne s'est pas déroulée comme prévu		
<input type="checkbox"/>	Sans avis		
Anomalies constatées :			
Commentaires :			

# Annexe C Cahier d'évaluation Cognitive Walkthrough

## 1-Introduction

### 1.1-Objectif du document

Ce document présente les cas de test de l'environnement CWE conformément aux spécifications de test décrit dans le Plan d'évaluation. Elle doit réaliser les scénarii décrits dans la section 3 ci-dessous, sur un scénario issu du site de la SNCF.

### 1.2-Références

Documents de référence

Plan d'évaluation

Fiche d'évaluations

Fiche problème

## 2-Équipe de test

Fonction	Nom et signature
Chef de projet	Thomas MAHATODY
Exécutant	Christophe KOLSKI
Exécutant	Mouldi Sagar
Exécutant	Kathia Oliveira

### 3-Scenarii

Cas de test	CT-FONC-01			
Titre	Spécifier le profil de l'évaluateur			
Objectif	Vérifier si le système permet de spécifier le profil de l'évaluateur			
Exigence de test	Spécifier le profil de l'évaluateur			
Rôle	Superviseur ou Inspecteur			
Exécutable	cwe.jar			
Procédure de test				
Précondition	Système lancé			
N°	Action	Entrée	Comportement attendu	OK?
1	Cliquer sur le menu « Fichier »		Un menu contenant « Profil utilisateur » se déroule	
2	Cliquer sur « Profil utilisateur »		L'écran « Écran d'identification utilisateur » s'affiche	
3	Cliquer sur l'onglet « Identification »		Le formulaire « Information d'identification » s'affiche	
4	Remplir le formulaire			
5	Cliquer sur le bouton « Enregistrer »		Un message « Login info enregistrés » s'affiche	
6	Cliquer sur le bouton « OK »		Le message disparaît	
7	Cliquer sur l'onglet « Connaissance Informatique »		Le formulaire « Expérience en informatique » s'affiche	
8	Remplir le formulaire			
9	Cliquer sur le bouton « Enregistrer »		Un message « Connaissances informatiques enregistrés » s'affiche	
10	Cliquer sur le bouton « OK »		Le message disparaît	
11	Cliquer sur l'onglet « Expérience en CW »		Le formulaire « Expérience en CW » s'affiche	
12	Remplir le formulaire			
13	Cliquer sur le bouton « Enregistrer »		Un message « Expérience en CW enregistrés » s'affiche	
14	Cliquer sur le bouton « OK »		Le message disparaît	
15	Cliquer sur l'onglet « Expérience en CWE »		Le formulaire « Expérience en CWE » s'affiche	
16	Remplir le formulaire			
17	Cliquer sur le bouton « Enregistrer »		Un message « Expérience en CWE enregistrés » s'affiche	
18	Cliquer sur le bouton « OK »		Le message disparaît	

Annexe

Cas de test	CT-FONC-02
Titre	Décrire le système à évaluer
Objectif	Vérifier si le système permet de décrire le système à évaluer
Exigence de test	Décrire le système à évaluer
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Spécification » se déroule	
2	Cliquer sur « Spécification »	L'écran « Écran de description de système » s'affiche	
3	Cliquer sur l'onglet « Système »	Le formulaire de description du système s'affiche	
4	Cliquer sur « Créer »	L'onglet « Identification » est initialisé	
5	Saisir le nom et la description du système		
6	Choisir le domaine dans la liste ou Saisir le nom de domaine et sa description puis cliquer sur le bouton « Ajouter ce domaine »	Le nouveau domaine apparaît dans la liste des domaine si on a cliqué sur le bouton « Ajouter ce domaine »	
7	Choisir la motivation du système <sup>8</sup>		
8	Choisir le contexte d'utilisation du système <sup>9</sup>		
9	Saisir un rôle		
10	Cliquer sur le bouton « Ajouter un rôle »	Le nouveau rôle apparaît dans la liste des rôles	
11	Cliquer sur le bouton « Ajouter une fonctionnalité »	La fonctionnalité « Nouvelle fonctionnalité » apparaît dans la liste des fonctionnalités	
12	Sélectionner la « Nouvelle fonctionnalité »	La fonctionnalité sélectionnée apparaît dans le formulaire à droite de la liste	
13	Modifier la nouvelle fonctionnalité		
14	Cliquer sur le bouton « OK »	La nouvelle fonctionnalité est modifié dans la liste des fonctionnalités	
15	Répéter 13 et 14 jusqu'à ce qu'on n'a plus de fonctionnalité à ajouter		
16	Cliquer sur le bouton « Enregistrer »	Un message « Expérience en CWE enregistré » s'affiche	
17	Cliquer sur le bouton « OK »	Le message disparaît	

<sup>8</sup>Deux classes de motivation : la motivation industrielle et commerciale et la motivation domestique.

<sup>9</sup>Les 3 classes de contexte sont : le contexte embarqué, le contexte portable et les autres qui ne sont pas ni embarqué ni portable. Le contexte embarqué concerne les interfaces intégrées ou incorporées avec le dispositif. C'est le cas des appareils domestiques comme les fours, les machines à laver. Quant au contexte portable, il concerne les interfaces accessibles via des dispositifs portables.

Annexe

Cas de test	CT-FONC-03
Titre	Décrire les connaissances des utilisateurs concernant le système
Objectif	Vérifier si le système permet de décrire les connaissances des utilisateurs le système à évaluer
Exigence de test	Décrire les connaissances des utilisateurs concernant le système
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Modélisation utilisateur se déroule	
2	Cliquer sur « Modélisation utilisateur »	L'écran « Écran d'identification utilisateur » s'affiche	
3	Cliquer sur l'onglet « Identité »	Le formulaire « Identité de l'utilisateur » s'affiche	
4	Saisir le nom du système		
5	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, la description est affichée et le formulaire identité de l'utilisateur est initialisé	
6	Cliquer sur le bouton « ajouter utilisateur »	Un utilisateur « Nouveau utilisateur » s'affiche dans la liste des utilisateurs	
7	Sélectionner « Nouveau utilisateur » dans la liste	« Nouveau utilisateur » s'affiche dans le formulaire identité de l'utilisateur	
8	Modifier le nouveau utilisateur dans le formulaire		
9	Cliquer sur le bouton « Enregistrer »	Le nouveau utilisateur est mis à jour dans la liste	
10	Répéter 6-9 jusqu'à ce qu'on n'a plus d'utilisateur à ajouter		

Annexe

Cas de test	CT-FONC-04
Titre	Choisir les scenarii à évaluer
Objectif	Vérifier si le système permet de décrire les scenarii à évaluer
Exigence de test	Choisir les scenarii à évaluer
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Choix de scenarii » se déroule	
	Cliquer sur « Choix de scenarii »	L'écran « Écran de description de scénario » s'affiche	
2	Cliquer sur l'onglet « Description scénario »	Le formulaire « Description de scénario » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, la description est affichée, les fonctionnalités du système s'affichent dans l'entête de la liste des scenarii	
5	Cliquer sur le bouton « ajouter scénario »	Un scénario « Nouveau scénario » s'affiche dans la liste des scenarii	
6	Sélectionner le nouveau scénario	Le nouveau scénario apparaît dans le formulaire en bas de la liste de scénario	
7	Modifier le nouveau scénario		
8	Choisir les acteurs du scénario en sélectionnant un acteur puis en cliquant sur le bouton « > »	L'acteur sélectionné passe de la liste des acteurs vers la liste des acteurs choisis	
9	Cliquer sur le bouton « Valider »	Le nouveau scénario est mis à jour dans la liste	
10	Dans les colonnes fonctionnalités Saisir « Oui » si le scénario concerne la fonctionnalité.		
11	Répéter 5-10 jusqu'à ce que l'ensemble de scénario couvre le maximum de fonctionnalités		
12	Cliquer sur le bouton « Enregistrer les scenarii »	Un message « Scenarii enregistrés » s'affiche	
12	Cliquer sur le bouton « OK »	Le message disparaît	

Annexe

Cas de test	CT-FONC-05
Titre	Analyser les tâches
Objectif	Vérifier si le système permet de décomposer les scenarii en suite d'action
Exigence de test	Analyser les tâches
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Analyse de tâches » se déroule	
2	Cliquer sur « Analyse de tâches »	L'écran « Écran d'analyse de tâche » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, les scenarii apparaissent dans la liste déroulante	
5	Choisir un scénario dans la liste déroulante		
6	Cliquer sur un scénario ou tâche ou sous tâche dans la liste des tâches	L'objet sélectionné apparaît dans le formulaire de droite	
7	Remplir le formulaire de droite		
8	Cliquer sur valider	L'objet modifié est mis à jour dans la liste des tâches	
9	Décomposer l'objet sélectionner en ajoutant d'objet fils (ajouter une tâche fille, sous-tâche fille, sous , etc.)	Un nouveau objet fils apparaît dans la liste hiérarchisée.	
10	Répéter 6-9 jusqu'à ce que tout est décomposé en action		
11	Cliquer sur le bouton « Enregistrer »	Un message « Scénario enregistré » s'affiche	
12	Cliquer sur le bouton « OK »	Le message disparaît	
13	Reprendre à partir de 5 jusqu'à ce tous les scenarii sont décomposés		



Annexe

Cas de test	CT-FONC-06
Titre	Choisir les méthodes d'évaluation
Objectif	Vérifier si le système permet de choisir une méthode d'évaluation et de l'affecter à une tâche et un évaluateur
Exigence de test	Choisir les méthodes d'évaluation
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Préparation »	Un menu contenant « Analyse de tâches » se déroule	
2	Cliquer sur « Analyse de tâches »	L'écran « Écran d'analyse de tâche » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, les scénarii apparaissent dans la liste déroulante	
5	Choisir un scénario dans la liste déroulante	La description du scénario apparaît. Les tâches et les sous-tâches apparaissent dans la liste des sous-tâches	
6	Sélectionner une sous-tâche		
7	Cliquer sur la partie évaluateur pour afficher la liste déroulante des évaluateurs	La liste déroulante des évaluateurs apparaît	
8	Choisir un évaluateur		
9	Cliquer sur la partie méthode pour afficher la liste déroulante des méthodes	La liste déroulante des méthodes apparaît	
10	Choisir une méthode		
11	Cliquer sur le bouton « Ajouter une session » si on veut ajouter une autre session d'évaluation	Une nouvelle ligne correspondant à la sous-tâche sélectionnée apparaît	
12	Répéter 6-11 jusqu'à ce que qu'on a le nombre de session souhaité pour chaque sous-tâche		
13	Cliquer sur le bouton « Enregistrer »	Un message « Scénario enregistré » s'affiche	
14	Cliquer sur le bouton « OK »	Le message disparaît	

Annexe

Cas de test	CT-FONC-07
Titre	Inspecter les tâches
Objectif	Vérifier si le système permet d'inspecter les tâches
Exigence de test	Inspecter les tâches
Rôle	Inspecteur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Inspection »	Un menu contenant « Évaluer » se déroule	
2	Cliquer sur « Évaluer »	L'écran « Écran d'inspection » s'affiche et le nom de l'évaluateur connecté ainsi que les tâches qu'il doit évaluer s'affichent aussi	
5	Choisir une tâche dans la liste déroulante	La première action et la première action apparaissent	
6	Choisir la réponse et saisir la justification		
7	Si la réponse est non, remplir le formulaire de problème d'utilisabilité		
8	Cliquer sur le bouton « Ajouter » si on veut ajouter un autre problème	Le formulaire de problème est initialisé	
9	Remplir le formulaire de problème d'utilisabilité		
10	Répéter 8-9 jusqu'à ce qu'on n'a plus de problème		
11	Cliquer sur le bouton « Question suivante »		
12	S'il y a encore de question, reprendre à partir de 6		
	S'il n'y a plus de question, cliquer sur le bouton « Action suivante »		
	Reprendre à partir de 6		
	Cliquer sur le bouton « Terminer »		
	S'il y a encore de tâche, reprendre à 5		

Annexe

Cas de test	CT-FONC-08
Titre	Calculer les tableaux de bord
Objectif	Vérifier si le système permet de calculer les tableaux de bord après l'inspection
Exigence de test	Calculer les tableaux de bord
Rôle	Superviseur
Exécutable	cwe.jar
Procédure de test	
Précondition	Système lancé

N°	Action	Comportement attendu	OK?
1	Cliquer sur le menu « Analyse »	Un menu contenant « Calculer tableau de bord » se déroule	
2	Cliquer sur « Calculer tableau de bord »	L'écran « Écran tableau de bord résultat » s'affiche	
3	Saisir le nom du système		
4	Cliquer sur le bouton « Chercher » ou taper la touche « entrée »	Si le système existe, les scénarii apparaissent dans la liste déroulante	
5	Choisir un scénario dans la liste déroulante	La description du scénario apparaît. Les sous-tâches évaluées ainsi que les méthodes et les évaluateurs apparaissent chacune dans une liste déroulante	
8	Choisir une sous tâches, une méthode et un évaluateur	La liste des problèmes apparaît	
9	Cliquer sur l'onglet « Synthèse des problèmes par catégorie » pour voir la synthèse de problème par catégorie avec le graphique	La synthèse des problèmes par catégorie apparaît	
10	Cliquer sur l'onglet « Synthèse des problèmes par évaluateur » pour voir la synthèse de problème par évaluateur avec le graphique	La synthèse des problèmes par évaluateur apparaît	

## Annexe D Fiche d'évaluation

Référence du scénario :

Date :

Référence de l'action :

Nom de l'évaluateur :

Questions	Réponse	Justification
Est-ce que l'utilisateur tente de réaliser l'effet désiré ?	<input type="checkbox"/> -Oui/ <input type="checkbox"/> Non	
	N° Problèmes	
Est-ce que l'utilisateur constate que l'action correcte est disponible ?	<input type="checkbox"/> -Oui/ <input type="checkbox"/> Non	
	N° Problèmes	
Est-ce que l'utilisateur associe l'action correcte avec l'effet qu'il tente de réaliser ?	<input type="checkbox"/> -Oui/ <input type="checkbox"/> Non	
	N° Problèmes	
Si l'action correcte est exécutée, est ce que l'utilisateur se rend compte qu'il progresse vers la réalisation de la tâche ?	<input type="checkbox"/> -Oui/ <input type="checkbox"/> Non	
	N° Problèmes	

---

## Annexe E Fiche problèmes

Référence du scénario :		Date :
Référence de l'action :		Nom de l'évaluateur :
Référence de la question :		Numéro du problème :
<b>Description :</b>		
<b>Justification :</b>		
<b>Fréquence</b>		<b>Sévérité</b>
☐Rarement ☐Souvent ☐Très souvent ☐Toujours		☐Tolérable ☐Modéré ☐Sérieux ☐ Critique
<b>Pourcentage</b>		<b>Catégorie</b>
☐0 ☐25 ☐50 ☐75 ☐100		☐Utilisateur ☐Affichage ☐Retour d'information ☐Séquence d'action ☐Texte&Icône ☐Autre
<b>Recommandation :</b>		

## **Spécification et conception d'un environnement d'aide à l'inspection des Systèmes Interactifs, basé sur la méthode Cognitive Walkthrough**

Cette thèse s'intéresse à l'évaluation des systèmes interactifs. Elle vise à contribuer à l'extension et à l'outillage de la méthode Cognitive Walkthrough. Cette méthode a fait l'objet de nombreuses études et expérimentations au niveau international. Elle a aussi conduit à plusieurs modifications et extensions. Malgré l'intérêt que suscite la méthode, son utilisation pose de nombreux problèmes aussi bien méthodologiques qu'applicatifs. Dans un premier temps, cette thèse décrit les éléments clés liées à la prise en compte des facteurs humains dans la conception et l'évaluation des systèmes interactifs. Puis, elle propose un état de l'art sur l'évaluation des systèmes interactifs basée sur la méthode Cognitive Walkthrough. Ensuite la spécification d'un environnement d'aide à l'inspection des systèmes interactifs basé sur la méthode Cognitive Walkthrough est décrite. Une maquette de l'environnement dénommé CWE (« Cognitive Walkthrough Environment ») est proposée ; elle a fait l'objet d'une première évaluation. Une conclusion et des perspectives terminent le mémoire.

Mots clés : Interaction Homme-Machine, Evaluation, Cognitive Walkthrough, Environnement d'évaluation.

## **Specification and design of an environment for interactive system inspection based on the Cognitive Walkthrough Method**

This thesis is focused on the evaluation of interactive systems. It aims to contribute to the extension and equipment of the Cognitive Walkthrough method. This method is the subject of many studies and experiments worldwide. It has also undergone several modifications and extensions. Despite the interest in the method, its use poses many methodological and applicative problems. Firstly, this thesis describes the key elements related to human factors consideration in interactive system design and evaluation. Then, it proposes a state of the art on the evaluation of interactive systems based on the Cognitive Walkthrough method. Then the specification of an environment to support the inspection of interactive systems based on Cognitive Walkthrough is described. A mockup of an environment called CWE ("Cognitive Walkthrough Environment") is proposed; a first evaluation is exposed. A conclusion and perspectives end the thesis.

Keywords: Human-Computer Interaction, Evaluation, Cognitive Walkthrough, Evaluation Environment.