



**HAL**  
open science

# Les COMETS : Une nouvelle Génération d'Interacteurs pour la Plasticité des Interfaces Homme-Machine

Olfa Daassi

► **To cite this version:**

Olfa Daassi. Les COMETS : Une nouvelle Génération d'Interacteurs pour la Plasticité des Interfaces Homme-Machine. Interface homme-machine [cs.HC]. UNIVERSITE JOSEPH-FOURIER GRENOBLE I, 2007. Français. NNT: . tel-01325453

**HAL Id: tel-01325453**

**<https://auf.hal.science/tel-01325453v1>**

Submitted on 3 Jun 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.

**THESE**  
présentée par

**Olfa Dâassi**

pour obtenir le titre de  
DOCTEUR DE L'UNIVERSITE JOSEPH-FOURIER – GRENOBLE I  
(arrêtés ministériels du 5 juillet 1984 et du 30 mars 1992)  
Spécialité Informatique

**Les COMETS : une nouvelle génération d'Interacteurs pour la  
Plasticité des Interfaces Homme-Machine**

Thèse soutenue le : 30 Janvier 2007  
devant le jury composé de :

Président : C. Garbay, Directeur de Recherches au CNRS, Grenoble  
Rapporteurs : C. Kolski, Professeur à l'Université de Valenciennes  
J. Vanderdonckt, Professeur à l'Université catholique de Louvain  
Examineurs : F. Tarpin-Bernard, Maître de Conférences à l'INSA de Lyon  
G. Calvary, Maître de Conférences à l'UJF, Directeur de thèse  
J. Coutaz, Professeur à l'UJF, Directeur de thèse

Thèse préparée au sein du laboratoire CLIPS  
(Communication Langagière et Interaction Personne-Système)  
de l'institut IMAG  
(Informatique et Mathématiques Appliquées de Grenoble)



A mes parents  
A ma famille  
A mes amis

## Remerciements

Je tiens à remercier :

Gaëlle Calvary qui m'a encadrée tout au long de ce travail et pour avoir su diriger cette thèse. Je la remercie pour ses encouragements en continue, pour ses valeurs scientifiques et humaines, Pour tout cela je la remercie vivement et sûrement plus,

Joëlle Coutaz qui m'a accueilli dans son équipe de recherche et qui m'a fait découvrir et aimer le domaine de l'interaction Homme-Machine,

Catherine Garbay, directrice du laboratoire CLIPS d'avoir accepté de présider le jury,

Jean Vanderdonckt et Christophe Kolski qui ont bien accepté d'être rapporteurs de ce document. Je les remercie très sincèrement pour leur intérêt et leurs critiques constructives,

Franck Tarpin-Bernard qui a accepté d'être membre du jury.

Je remercie également :

Tous les membres de l'équipe IIHM qui ont contribué à créer ce cadre de travail coopératif et amical,

Tous les membres du projet européen Caméléon.

Chaouki (mon frère) pour ses conseils durant ces années d'études universitaires.

Sana, Mohamed et Dima pour leur soutien et encouragements.

Et à toute ma famille et surtout mes parents, je dis Merci.

## TABLE DES MATIERES

---

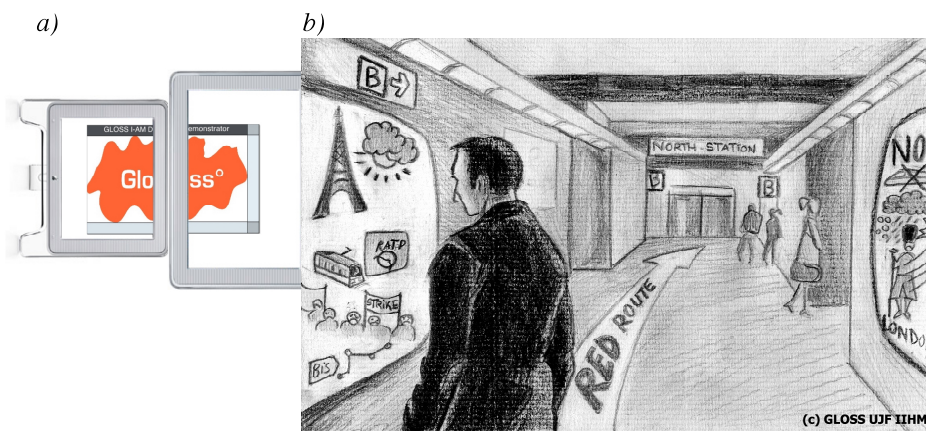
<b>Introduction .....</b>	<b>8</b>
1. Le problème : la plasticité .....	11
2. Le sujet : des interacteurs pour la plasticité .....	12
3. Les objectifs de la thèse .....	14
4. L'approche .....	14
5. Le plan du mémoire .....	15
<b>Chapitre I.....</b>	<b>.....</b>
<b>Les Interacteurs : définition et exigences pour la plasticité .....</b>	<b>16</b>
1. Définition et concepts de base.....	17
2. Vision des interacteurs pour la plasticité .....	24
3. Exigences pour la plasticité.....	28
4. Profil minimal requis.....	32
<b>Chapitre II .....</b>	<b>.....</b>
<b>Etat de l'art des interacteurs pour la plasticité : Revue et caractérisation .....</b>	<b>34</b>
1. Prédominance fonctionnelle.....	35
1.1 Fonction métier .....	36
1.2 Présentation .....	38
2. Prédominance ontologique.....	44
2.1 Etat .....	45
2.2 Structure .....	45
3. Prédominance génétique .....	48
3.1 Détection du changement de contexte d'usage .....	48
3.2 Calcul et mise en œuvre de réaction .....	49
4. Synthèse .....	59
<b>Chapitre III.....</b>	<b>.....</b>
<b>Apports pluridisciplinaires pour la valeur .....</b>	<b>62</b>
1. Cadre de référence révisé pour la Plasticité des IHM .....	62
2. Ressources .....	66
2.1 Espace de classification.....	66
2.2 Apports pluridisciplinaires .....	67
2.2.1 Métier .....	67
Perspective utilisateur .....	68
Perspective système .....	76
Perspective environnementale.....	78
2.2.2 Adaptation .....	78
3. Propriétés.....	80
3.1 Espace de classification.....	80
3.2 Apports pluridisciplinaires .....	80
3.2.1 Métier .....	80
Perspective fonctionnelle .....	80
Perspective non fonctionnelle .....	81

3.2.2	Adaptation .....	84
	Perspective fonctionnelle .....	84
	Perspective non fonctionnelle .....	85
4.	Synthèse .....	87
<b>Chapitre IV .....</b>		
<b>Les Comets .....</b>		<b>90</b>
1.	Définition .....	90
2.	Classification .....	91
2.1	Autodescription .....	91
2.2	Adaptation .....	92
3.	Modélisation .....	93
3.1	Description multifacettes .....	94
3.2	Transformations et propriétés .....	98
3.3	Transformations et ressources .....	101
3.3.1	Ressources physiques .....	103
3.3.2	Ressources numériques .....	105
3.3.3	Ressources cognitives, conatives et temporelles .....	105
3.4	Synthèse .....	106
4.	Adaptation .....	107
4.1	Stratégies .....	107
4.1.1	Stratégies de constitution .....	107
4.1.2	Stratégies de valeur .....	108
4.2	Politiques .....	108
5.	Architecture logicielle : Compact .....	109
6.	Synthèse .....	110
<b>Chapitre V .....</b>		
<b>Les Comets en action : demonstreur decor .....</b>		<b>112</b>
1.	Cas d'étude .....	112
2.	COMETs .....	118
2.1	COMETs IHM .....	118
2.2	COMETs Méta-IHM .....	123
3.	Architecture logicielle .....	124
4.	Synthèse .....	125
<b>Conclusion et perspectives .....</b>		<b>126</b>
1.	Contributions .....	126
2.	Perspectives .....	127
<b>Bibliographie .....</b>		<b>130</b>

## INTRODUCTION

---

Avec les avancées des réseaux et les progrès en miniaturisation, les réseaux sans fil se généralisent et favorisent l'infiltration de dispositifs ultra-légers dans les entreprises et les foyers : l'informatique devient diffuse, reléguant l'ordinateur boîte grise à une vision du passé. Désormais, dans un avenir proche, l'assistant personnel (PDA) devient télécommande universelle ; nos objets quotidiens s'amplifient et deviennent supports possibles à l'interaction. Les dispositifs s'assemblent, se désassemblent de manière opportuniste, créant un espace interactif au service d'une information située. Par exemple, la juxtaposition de deux écrans fournit une plus large surface d'affichage (Figure 1a) ; la météo à Paris surgit sur le mur de l'aérogare au passage de Bob en partance pour Paris (Figure 1b).



*Figure 1 : Avec l'informatique ambiante, l'espace interactif supplante le traditionnel bureau (« desktop »). Les dispositifs s'assemblent, se désassemblent de manière opportuniste et fournissent une information située[.]*

Dans cette vision, l'hypothèse implicite de la plate-forme d'exécution unique et du lieu d'interaction fixe ne tient plus : l'utilisateur est mobile et met à profit, de façon opportuniste, son espace interactif pour accomplir sa tâche. Cette évolution de l'informatique métamorphose l'interaction Homme-Machine. Les Interfaces Homme-Machine (IHM) passent [Calvary 06] :



- Du *centralisé* au *distribué*. Pour exemples : la métaphore du peintre explorée dans Pick and Drop [Rekimoto 97] (Figure 2) ou la télécommande universelle développée dans Pebbles [Myers 01] ;

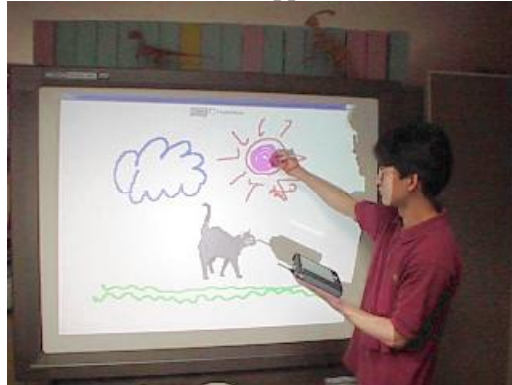


Figure 2 : La métaphore du peintre dans Pick and Drop. L'IHM de l'éditeur de dessin est distribuée entre le PDA et le PC. Image extraite de <http://www.csl.sony.co.jp/person/rekimoto/pickdrop/>.

- Du *classique* à l'*exotique* dans les dispositifs d'entrée et de sortie (Figure 3). Pour mémoire, le crayon communicant DigitalInk [Kasabach 98]; AmbientRoom [Ishii 98] qui utilise des fioles comme dispositifs de sortie ou plus récemment le lapin de Violet (le « Nabaztag ») qui allie entrée et sortie. On se souvient aussi de la fenêtre perceptuelle de F. Bérard qui imagine la tête comme dispositif d'entrée [Bérard 99] (a) et plus récemment les travaux de P. Antoniac qui voient en la main une surface d'affichage [Antoniac 02] (b) ;

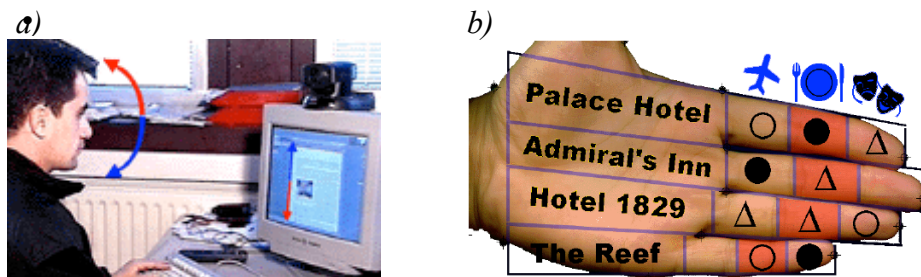


Figure 3 : Avec l'informatique ambiante, de nouveaux dispositifs d'entrée et de sortie sont imaginés. Le corps humain, en particulier, peut être vu comme un vivier de ressources d'interaction en entrée (a) comme en sortie (b). Images respectivement extraites de [Bérard 99] et [Antoniac 02].

- Du *monomodal* au *multimodal*. Classiquement, l'interaction Homme-Machine privilégie la vue, le toucher et l'ouïe. L'olfactif est désormais à l'étude pour le plaisir du nez et la sécurité. Dans le projet

Exhalia, par exemple, de France Telecom, un diffuseur de fragrances est connecté à un PC (Figure 4). Lorsque l'utilisateur navigue sur Internet, les images sont accompagnées d'odeurs : c'est le concept du "web olfactif". Pour la sécurité, l'idée serait par exemple d'exprimer un danger ou d'en renforcer le signal par une odeur diffuse. L'utilisateur n'est alors plus rivé devant l'écran ou le voyant. L'odeur est diffusée là où il se trouve, satisfaisant ainsi l'exigence de qualité des messages d'erreur telle qu'exprimée dans le référentiel de critères ergonomiques de C. Bastien et D. Scapin [Bastien 93]. A noter que cette idée n'est autre que celle déjà appliquée dans la vie réelle : le gaz est, par nature, inodore. Son parfum lui est artificiellement conféré, à seule visée de détection ;



Figure 4 : Un diffuseur de fragrances pour le web olfactif.  
Actualité du 16/06/2005 extraite de  
<http://www.les1000delouest.com/actualites/>.

- Du *sédentaire* au *nomade*. Jusqu'ici "scotchées" à leur ordinateur d'exécution, les IHM "valsent" désormais dans leur espace interactif, au gré de l'utilisateur, selon l'arrivée et le départ de ressources. Elles migrent partiellement ou totalement, s'adaptant aux capacités de la plate-forme cible. Typiquement, dans les surfaces augmentées de Rekimoto [Rekimoto 99], la présentation des objets (tables et chaises) s'adapte à l'inclinaison horizontale (vue de dessus 2D – Figure 5a) ou verticale (perspective 3D – Figure 5b) de la surface d'affichage.

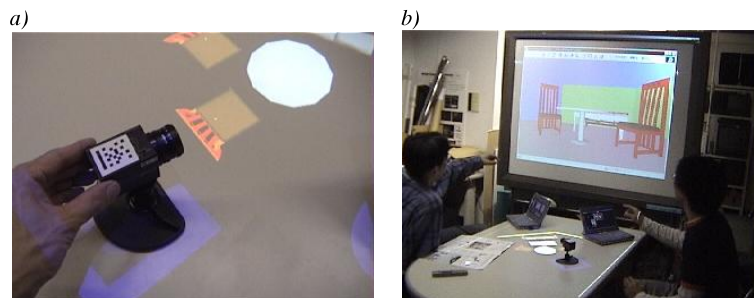


Figure 5 : Selon l'inclinaison de la surface d'affichage, les objets sont représentés en vue de dessus (a) ou perspective (b) [Rekimoto99].

Si cette vision séduit du point de vue de l'usage, elle peut effrayer par la complexité à l'apparence "sans limite" de son ingénierie. Les méthodes et outils actuels n'ont, en effet, pas été conçus pour concevoir, développer et évaluer de telles IHM. Cette thèse traite de l'ingénierie d'IHM avancées. La voie explorée est celle de la *plasticité*.

En introduction, la première section définit la propriété de plasticité. La deuxième section présente le sujet de ma thèse. Mes objectifs et mon approche font l'objet des sections 3 et 4. La section 5 annonce le plan de mon mémoire.

---

## 1. Le problème : la plasticité

En 1999, face au succès des ordinateurs de poche et à la diversité croissante des plates-formes, J. Coutaz et D. Thevenin concluent à l'inadéquation des approches au cas par cas. Ils introduisent une nouvelle propriété, la *plasticité*, pour dénoter la capacité d'une IHM à s'adapter à son contexte d'usage dans le respect de son utilisabilité [Thevenin 99]. Le contexte d'usage y est défini comme un triplet <utilisateur, plate-forme, environnement> où :

- L'utilisateur représente l'utilisateur final du système interactif. Il peut être décrit par des données générales (âge, taille, etc.), ses compétences dans le domaine applicatif et en informatique ;
- La plate-forme cerne les requis matériels et logiciels nécessaires à l'interaction. Typiquement, les dispositifs d'entrée, de sortie, de calcul et de communication ;
- L'environnement dénote l'espace physique hébergeant l'interaction. Il peut être décrit par ses conditions lumineuses, sonores, sociales, etc.

Depuis, deux angles d'attaque s'affirment selon que les travaux portent sur l'interaction située [Suchman 87] et la technologie support dite « context aware computing » [Dey 00, Cheverst 01] ou les IHM dites multi-plates-formes [Thevenin 01]. Ma thèse s'inscrit dans ce deuxième volet.

Les avancées en plasticité ont aujourd'hui permis d'en comprendre l'espace problème, de proposer des solutions partielles (méthodes, modèles et outils) sans pour autant doter les concepteurs d'un *atelier de plasticité*, c'est-à-dire d'un environnement de conception, d'exécution et d'évaluation d'IHM plastiques. La thèse se place selon cette perspective système : sa vocation est d'offrir aux concepteurs et développeurs d'IHM des outils appropriés pour l'ingénierie d'IHM plastiques. Les outils ici étudiés sont une nouvelle génération d'interacteurs : les *comets* (Context Mouldable widgETs).

## 2. Le sujet : des interacteurs pour la plasticité

Les interacteurs sont un angle d'attaque "chronique" en ingénierie d'IHM. On se souvient, en particulier, du projet européen AMODEUS qui amorçait une théorie des interacteurs. Dans ces travaux, certains adoptaient une perspective analytique [Duke 95], d'autres un point de vue d'ingénierie [Faconti 93]. Depuis, les travaux se multiplient. Citons, par exemple, P. Markopoulos en architecture logicielle [Markopoulos 97], J. Vanderdonck sur les niveaux d'abstraction [Vanderdonck 97] ou plus récemment D. Thevenin sur l'adaptation [Thevenin 01]. Nous nous situons en prolongement de ces travaux mais avec une ambition de couvrir non seulement la conception, mais aussi l'exécution et l'évaluation d'IHM plastiques. Il ne s'agit pas de prôner et produire des interacteurs auto-adaptatifs assumant en toute autonomie leur adaptation, mais d'étudier les exigences, au grain des interacteurs, pour faciliter l'ingénierie d'IHM plastiques.

Les widgets des boîtes à outils actuelles révèlent des prédispositions limitées pour l'adaptation :

- Les widgets se redimensionnent certes, mais sont incapables typiquement de changer de modalité dès lors que des dispositifs d'interaction viennent à manquer. Un bouton graphique, par exemple, ignore ses requis en termes de dispositifs d'entrée et de sortie et est incapable de se transformer en commande vocale dès lors que ces ressources ne sont plus disponibles. Il n'est pas capable non plus de se projeter sur la main dès lors que l'utilisateur souhaite être mobile ou que l'écran n'est plus disponible ;
- La raison d'être de chaque widget reste à un stade documentaire : ni l'IHM, ni ses éléments constitutifs ne connaissent la contribution de chaque widget à la tâche plus globale que réalise l'utilisateur. Aussi, toute transformation opportuniste de l'IHM sans préfabrication est-elle compromise : si la tâche « consulter date », par exemple, est optionnelle et la surface d'affichage restreinte, les widgets correspondant à cette tâche ne pourront pas être supprimés sans que l'informaticien ait prévu et codé l'IHM correspondante. C'est typiquement le cas dans FlexClock [Grolaux 01] (Figure 6) où l'apparition et la disparition de la date ont été prévues à la conception et codées "en dur" dans l'application ;

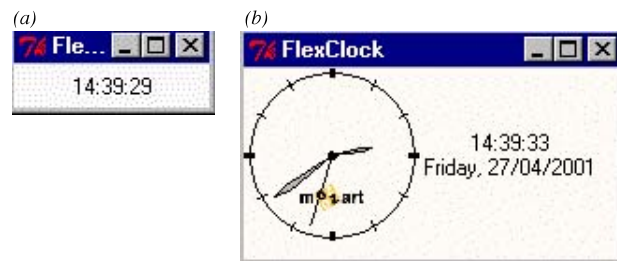


Figure 6 : FlexClock [Grolaux 01], un démonstrateur de plasticité. L'IHM se remodèle selon la taille de la fenêtre. L'apparition et la disparition de la date ont été prévues à la conception.

En conséquence, ce sont aujourd'hui les développeurs qui pallient l'insuffisance des outils. Les corollaires sont :

- Des coûts de développement et de maintenance élevés pour produire autant de versions d'IHM que de contextes d'usage pressentis à la conception ;
- Des incohérences ergonomiques potentielles entre ces versions. Pour exemple, les applications bancaires dont les versions "distributeur" et Internet sont incohérentes par quatre aspects (Figure 7) [Daassi 03] :
  - Les tâches ne sont pas les mêmes. En particulier, le retrait d'argent est impossible sur Internet ;
  - L'agencement des tâches varie, intervertissant entre les deux plates-formes les tâches de virement et de consultation des comptes ;
  - La formulation des tâches diffère, variant de « Soldes des comptes » à « VOS COMPTES » ;
  - L'incarnation des tâches est différente, en particulier dans son couplage entre les mondes physique et numérique. Alors qu'en b), un seul bouton physique suffit (celui de la souris) à l'activation de tout service (« VOS COMPTES» ou « VIREMENTS »), en a) huit boutons sont requis (deux colonnes de quatre). Un bouton correspond à une tâche, mais n'est pas gravé du nom de cette tâche, les pages d'écran se succédant au fil de l'interaction. Aussi, des traits blancs sont-ils physiquement rajoutés pour forcer la perception du couplage entre les mondes physique (boutons) et numérique (libellés). En b), ce couplage est rendu observable par le pointeur souris. Lorsque ce pointeur arrive sur le libellé d'une tâche, il en permet l'activation. Cette balance entre les mondes physique et numérique se retrouve pour l'identification de l'utilisateur. Alors qu'elle est véhiculée par la carte de crédit en a), elle est assurée par un code virtuel en b).

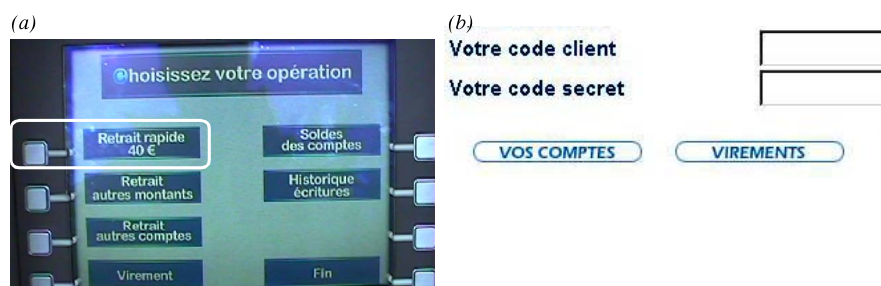


Figure 7 : Opérations bancaires sur distributeur (a) vs Internet (b). On constate que les IHM varient en termes de services, d'agencement, de formulation et d'incarnation. On notera en a) l'ajout de traits blancs pour matérialiser le couplage entre les mondes physique et numérique.

- Une incapacité à couvrir les contextes d'usage non prévus à la conception, en conséquence, à donner corps à cet opportunisme tant mis en avant dans les scénarii d'informatique ambiante.

Il convient, dans ces conditions, de réviser les acquis en interacteurs pour couvrir les requis de la plasticité.

---

### 3. Les objectifs de la thèse

Les objectifs de la thèse sont de :

- Descendre au grain des interacteurs des problèmes d'adaptation aujourd'hui réglés, faute d'outils, au niveau des systèmes interactifs. C'est le cas de FlexClock [Grolaux 01] typiquement ;
- Ainsi, capitaliser des "solutions" d'adaptation réutilisables et, en conséquence, favoriser la réalisation et donc la diffusion d'IHM plastiques ;
- Ouvrir les interacteurs sur l'opportunisme en leur permettant de :
  - Profiter de nouvelles ressources physiques et/ou numériques nouvellement apparues ;
  - Surmonter le départ de ressources physiques et/ou numériques.

Ceci suppose de pouvoir, à l'exécution, raisonner sur la raison d'être des interacteurs, leurs requis et leur qualité. J'espère ainsi, non seulement, faciliter (et donc encourager) l'ingénierie et l'exploration d'IHM plastiques mais aussi améliorer l'ingénierie traditionnelle par une explicitation d'un savoir-faire humain resté jusqu'ici bien souvent mental.

L'approche développée pour atteindre ces objectifs fait l'objet de la section suivante.

---

### 4. L'approche

Mon approche se veut analytique, étudiant ce qui :

- D'une part, dans la plasticité d'un système interactif peut être assuré au grain de l'interacteur ;
- D'autre part, dans un interacteur, peut contribuer à la plasticité des systèmes interactifs.

Cette approche m'incite, en premier lieu, à redéfinir le concept d'interacteur puis ériger un espace problème. Cet espace problème guidera mon état de l'art en matières d'interacteur. Les lacunes seront, si possible, comblées par des apports pluridisciplinaires ouvrant ainsi la thèse sur :

- D'autres problématiques en IHM, telles que la multimodalité, par exemple, qui sera ici vue comme un moyen d'adaptation ;
- D'autres disciplines comme le Génie Logiciel via l'Ingénierie Dirigée par les Modèles (IDM ou MDE pour Model Driven Engineering), la théorie des systèmes, ou encore l'économie via l'analyse de la valeur.

Ces éclairages nous conduisent à une nouvelle classification et modélisation d'interacteurs. Nous mettons en oeuvre cette proposition conceptuelle sur un cas d'étude : DECOR (DEmonstrateur de COMets en Remodelage). Le plan de ce rapport suit cette approche.



## **5. Le plan du mémoire**

Le mémoire se structure en cinq chapitres :

- Le chapitre 1 pose une nouvelle définition des interacteurs et énonce des exigences pour la plasticité ;
- Le chapitre 2 examine l'état de l'art sur la base de la grille d'analyse établie au chapitre 1 ;
- Le chapitre 3 recense des apports pluridisciplinaires susceptibles de combler les manques révélés dans l'analyse critique ;
- Le chapitre 4 introduit une nouvelle génération d'interacteurs, les COMETS, dirigée par les exigences et mettant à profit les différents acquis ;
- Le chapitre 5 en donne une mise en oeuvre.

La thèse s'achève sur un rappel des contributions et l'ouverture sur de nombreuses perspectives.

# CHAPITRE I

## LES INTERACTEURS : DEFINITION ET EXIGENCES POUR LA PLASTICITE

---

La notion d'« interacteur » a été introduite par G. Faconti et F. Paterno en 1990 comme étant l'*abstraction* d'une *entité* capable à la fois d'entrée et de sortie dans un système interactif *graphique* [Faconti 90]. Cette définition est intéressante par les interrogations qu'elle pose :

- La notion d'abstraction. Très vite, une distinction est faite entre interacteurs abstraits et concrets [Morin 90]. La première implémentation en est donnée par J. Vanderdonck et al. en 1993 [Vanderdonck 93]. Elle sépare la fonction de l'interacteur (la tâche utilisateur qu'il offre, par exemple la sélection d'une option parmi N) de sa présentation (par exemple, les boutons radio ou listes déroulantes).
- La granularité. La notion d'entité reste vague. C'est un élément constitutif d'un système interactif sans qu'une taille ou une fonction n'en soit définie. Seuls ses flux d'entrée/sortie sont considérés. Une fenêtre de confirmation, par exemple, est-elle un interacteur ?
- L'interaction. Les entrées et sorties font-elles référence à l'interaction personne-système ou tout « composant » logiciel en interaction avec d'autres composants est-il interacteur ? Entrées et sorties sont-elles nécessaires pour être interacteur ou un libellé de consultation pure (pas d'entrée) est-il interacteur ?
- La modalité. Dès 1993, D. Duke et M. Harrison appellent à une généralisation de la définition pour être portée au-delà du graphique [Duke 93a]. Les interacteurs multimodaux de M. Crease [Crease 01] (son et graphique) vont dans ce sens.

Ces questions récurrentes dans la littérature appellent à une clarification de la définition. La première section y est consacrée. La deuxième section présente une vision des interacteurs pour la plasticité. Cette vision permet d'énoncer, en section 3, des exigences pour les interacteurs.

---



## 1. Définition et concepts de base

Les travaux sur les « interacteurs » sont nombreux mais la terminologie reste hésitante. Un amalgame est souvent fait entre « interacteur », « widget », « objet d'interaction »... Nous proposons ici une définition pragmatique permettant de raisonner sur la plasticité.

Un interacteur est une entité numérique offrant des fonctions de service, rendues observables et/ou manipulables à l'utilisateur par le biais d'entités physiques qui jouent le rôle de ressources d'interaction.

Dans cette définition, le terme de « fonction de service » est emprunté au monde des fonctions de J. Chevallier [Chevallier 89]. Les termes d'« entité numérique/physique » et « ressource d'interaction » sont conformes à l'ontologie multi-instrument multisurface de C. Lachenal [Lachenal 04]. Nous rappelons ici les définitions de ces notions clé : fonction de service, entité numérique/physique puis ressource d'interaction.

### ☛ Fonction de service

Telle que définie par J. Chevallier [Chevallier 89], une fonction est un intermédiaire entre un *besoin* de l'utilisateur et un *produit*. Dans l'exemple de l'auteur, si l'utilisateur souhaite voir clair pour se déplacer sans risque, son besoin n'est pas d'avoir une lampe mais d'avoir de la lumière. La fonction attendue est de produire un flux lumineux. La lampe est un produit tout désigné pour le faire. Ainsi, les fonctions sont-elles vues comme une réponse à un besoin de l'utilisateur dans un *milieu environnant*. Ce milieu environnant est l'ensemble des éléments matériels et immatériels entourant le produit pendant son utilisation, sa mise en œuvre, son entretien. Ce milieu peut exercer des actions appelées *contraintes*. Le produit doit y faire face.

J. Chevallier distingue les fonctions dites « de service » des fonctions « techniques » (Figure 8). Les *fonctions de service* sont une réponse à un besoin et à des contraintes, contrairement aux *fonctions techniques* (aussi dites élémentaires) qui sont internes au produit et résultent de choix du concepteur. Dans les fonctions de service, J. Chevallier distingue les fonctions principales, complémentaires et contraintes.

- Les fonctions principales sont la réponse directe au besoin de l'utilisateur. Elles répondent à un désir fondamental de l'utilisateur, à son exigence première. Elles correspondent au service rendu, par exemple, produire un flux lumineux. Ce sont ces fonctions principales qui motivent l'achat du produit. Ces fonctions peuvent être d'usage ou d'estime : d'usage, elles répondent alors à un besoin existentiel de l'utilisateur ; d'estime, elles augmentent son plaisir. La lampe de poche, par exemple, devra non seulement produire un flux lumineux mais aussi arborer de jolies formes et couleurs. L'esthétique, l'agrément, le silence, la sécurité, etc., relèvent de la fonction d'estime et contribuent au choix du produit. Dans certains

cas (typiquement les parfums ou les bouquets), ce sont ces fonctions d'estime qui dirigent l'achat du produit ;

- Les fonctions complémentaires ou secondaires ne sont pas directement liées à la satisfaction du besoin, mais y contribuent de façon périphérique. Par exemple, par des fonctions d'estime (être élégant pour un pull dont la fonction principale est de protéger du froid) ou des fonctions d'usage non essentielles (par exemple, ramasser l'herbe pour une tondeuse dont la fonction principale est de couper l'herbe ou encore indiquer l'état de charge des accumulateurs pour une lampe dont la fonction principale est de produire un flux lumineux) ;
- Les fonctions contraintes sont les réactions d'un produit face aux contraintes exercées par le milieu environnant. Pour la lampe, résister à la corrosion est une fonction contrainte qui répond à l'agressivité potentielle du milieu extérieur.

Contrairement aux fonctions de service que J. Chevallier définit comme l'enveloppe du produit ou encore sa zone d'échange avec ses utilisateurs, les fonctions techniques (aussi dites élémentaires) sont internes au produit. Par exemple, le contacteur qui assure la continuité électrique entre les accumulateurs et l'ampoule assure une fonction technique qui contribue à l'accomplissement de la fonction principale et donc de la satisfaction du besoin. La figure 8 modélise cette notion de fonction.

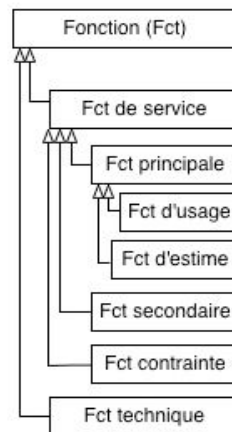


Figure 8 : Modélisation de la notion de fonction conformément à J. Chevallier [Chevallier 89].

### ☛ Entité numérique/physique

Les entités physiques sont au cœur de l'ontologie multi-instrument multisurface de C. Lachenal [Lachenal 04]. Dans ce travail, une situation d'interaction met en scène deux acteurs, l'utilisateur et le système, en symétrie par rapport à un ensemble d'entités physiques (Figure 9). Ils disposent chacun d'effecteurs pour agir sur ces entités physiques et de capteurs pour les observer.

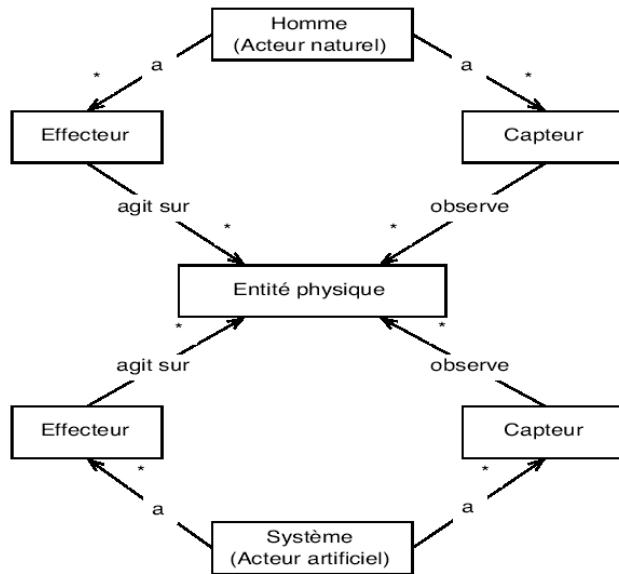


Figure 9 : Dans l'ontologie de C. Lachenal, les entités physiques sont au cœur de l'interaction homme-machine. Elles font l'objet d'observations et d'actions par l'utilisateur et le système via des capteurs et effecteurs [Lachenal 04].

Par exemple, dans le DigitalDesk [Wellner 93], le système suppose l'existence d'un vidéo projecteur (effecteur système) et d'une caméra (capteur système) orientés vers la feuille de dessin (entité physique). L'utilisateur perçoit le dessin de façon visuelle (ses capteurs sont ses yeux ; l'entité physique observée est la feuille) et le modifie à l'aide d'un crayon et d'une gomme (effecteurs humains).

Dans l'ontologie de C. Lachenal, les interacteurs sont des entités numériques rendant observables et manipulables à l'utilisateur l'information élaborée par le système (Figure 10). Leur observabilité résulte d'une action système de type projection, mobilisant le couplage d'effecteurs et d'entités physiques (par exemple, un vidéo projecteur orienté vers une table).

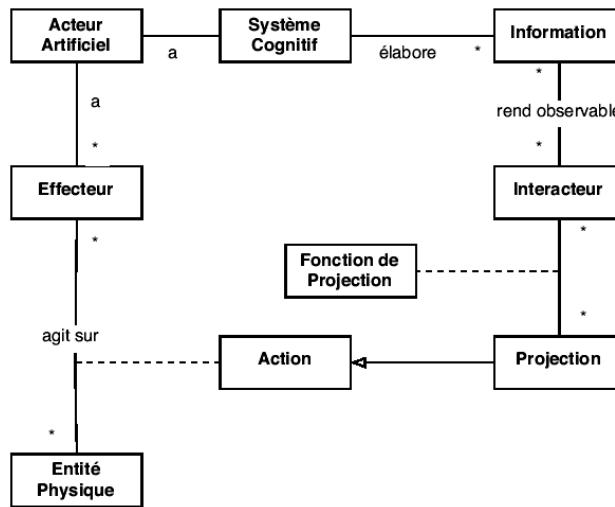


Figure 10 : Dans l'ontologie de C. Lachenal, les interacteurs sont l'objet de projection sur des entités physiques par l'usage d'effecteurs [Lachenal 04].

Dans cette ontologie, les entités physiques qui servent d'objets de médiation entre l'homme et le système sont dites "ressources d'interaction". Elles jouent un rôle d'interaction.

### ☛ Ressource d'interaction

C. Lachenal définit une ressource d'interaction comme une entité physique impliquée dans un "canal d'interaction", c'est-à-dire une séquence d'actions puis d'observation entre les acteurs humain et artificiel (se reporter à C. Lachenal pour une définition formelle). Une entité physique n'est alors pas intrinsèquement ressource d'interaction. Elle le devient dès lors qu'elle joue un "rôle d'interaction" : dispositif d'entrée (les instruments) ou de sortie (les surfaces) (Figure 11).

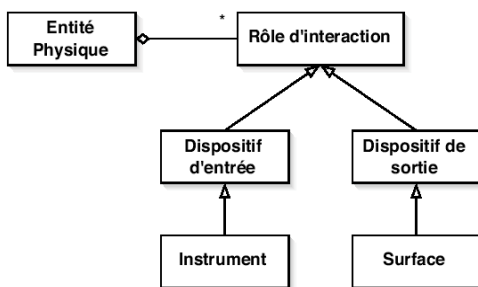


Figure 11 : Une entité physique est dite ressource d'interaction lorsqu'elle joue un rôle d'interaction. Extrait de C. Lachenal [Lachenal 04].

L'aptitude d'une entité à tenir un rôle d'interaction est dynamique, dépendant de ses caractéristiques et propriétés intrinsèques, mais aussi des capacités d'action et

d'observation des acteurs en présence. Ainsi, une entité qui, à un instant  $t$ , est ressource d'interaction, peut, à l'instant  $t+1$ , ne plus l'être et inversement. Un téléphone, par exemple, rend principalement la fonction d'appel. Mais dès lors qu'il est traqué par une caméra (capteur système), il peut jouer le rôle d'instrument devenant dispositif de pointage. Inversement, une main qui sert classiquement d'instrument peut servir de surface d'affichage dès lors qu'elle est perçue par le système et que celui-ci dispose d'un effecteur pour y projeter l'information (Figure 3b).

En conséquence, si "câbler" dans l'interacteur sa présentation avec l'hypothèse implicite de la disponibilité permanente des ressources d'interaction requises était une facilité jusqu'ici raisonnable, elle ne l'est plus désormais. La prolonger, c'est briguer l'opportunité de l'informatique ambiante et prendre le risque de priver l'utilisateur d'usages séduisants. Aussi, l'interacteur doit-il désormais se décrire en termes de fonctions de service et de ressources d'interaction requises. L'assignation entités physiques-rôles d'interaction sera dynamique, prise en charge par un gestionnaire de ressources, selon la "valeur" que l'utilisateur perçoit dans l'association.

En économie, la *valeur* est définie comme "une grandeur qui croît lorsque la satisfaction du besoin de l'utilisateur augmente ou que l'ensemble des dépenses diminue [Lachnitt 80]. La notion de valeur apparaît dès lors qu'un sujet, pour fixer son choix entre plusieurs produits, établit une relation subjective entre une impulsion (un besoin) et des freins (des dépenses)" [Chevallier 89]. La Figure 12 intègre cette notion de valeur : un interacteur représente une certaine valeur pour l'utilisateur pour un besoin donné. Cette valeur intègre bien les deux dimensions de l'économie à savoir :

- La satisfaction du besoin pour la contribution fonctionnelle de l'interacteur (association Interacteur-Fonction que nous étendons, sur la Figure 12, aux fonctions autres que de service) ;
- Le coût par l'utilisation d'entités numériques, physiques et cognitives pour les actions et observations humaines et système. L'interacteur est projeté par le système (action S) ce qui en permet l'observation par l'humain (observation H). Cette observation mobilise des capteurs qui informent le système cognitif de l'information perçue. Une spécification d'actions (action H) peut s'ensuivre. Les actions utilisateur sont observées par le système (observation S) ce qui permet l'exécution des fonctions de service correspondantes. Observations et actions requièrent un couplage entre entités physiques et capteurs ou effecteurs. Ce couplage est représenté dans la Figure 12 par les classes associatives Observation et Actions. Capteurs et effecteurs sont des entités physiques comme l'est aussi l'utilisateur. L'utilisateur possède ses propres capteurs et effecteurs (yeux, mains, etc.). A titre d'exemple, considérons le téléphone projeté sur la main (Figure 3b). Ce clavier est un assemblage de touches physiques. Considérons l'interacteur « touche », c'est-à-dire l'entité numérique dont les instances sont couplées aux entités physiques « touches ». Cet interacteur offre la fonction de service « spécifier un chiffre entre 0 et 9 ». Son affichage par le système

(action S) requiert le couplage entre un effecteur (ici le vidéo projecteur) et une entité physique sur laquelle se fait l'affichage (ici la main gauche, mais cela pourrait être un écran). Sa perception par l'utilisateur (observation H) requiert le couplage entre un capteur (ici les yeux de l'utilisateur compte tenu de l'affichage graphique) et cette même surface d'affichage (ici la main gauche - à noter que si c'était un écran, une contrainte apparaîtrait alors imposant que cette surface d'affichage soit à vue de l'utilisateur et lisible). La numérotation par l'utilisateur (action H) requiert le couplage entre un effecteur humain (ici la main droite) et l'entité physique sur laquelle s'effectue l'affichage (ici la main gauche). La perception par le système (observation S) de cette action humaine requiert le couplage d'un capteur (ici une caméra) et d'une surface d'affichage (ici la main gauche vers laquelle doit donc être orientée la caméra). Nous appellerons « ressources » ces entités physiques, numériques et cognitives requises par l'interaction.

En synthèse, notons le caractère intégrateur de la Figure 12. La modélisation recentre l'ontologie de C. Lachenal sur les interacteurs en intégrant :

- Le monde des fonctions ;
- Le cycle d'observations-actions tel que développé dans la Théorie de l'Action de Norman [Norman 86] ;
- Le modèle du Processeur Humain dans ses trois sous-systèmes perceptuel (capteurs), cognitif et moteur (effecteur) [Card 83] ;
- Et la notion de valeur telle qu'analysée en économie.

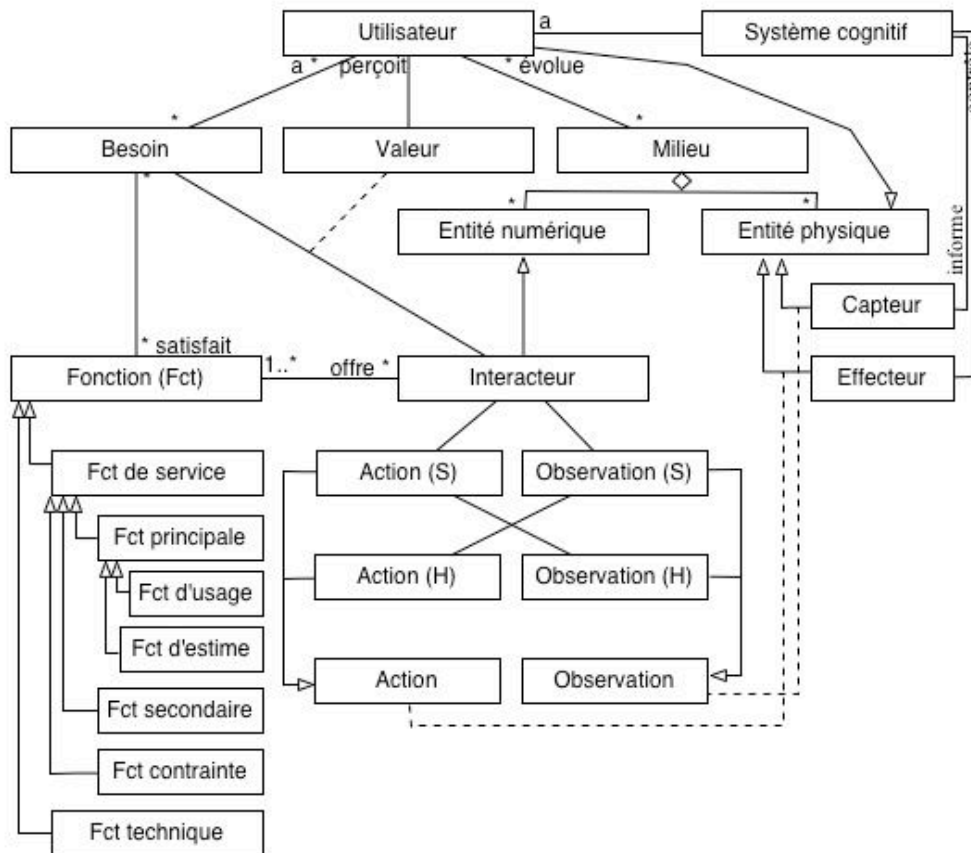


Figure 12 : Modélisation centrée "valeur" de la notion d'interacteur.

Avant de clore cette partie, nous répondons aux questions récurrentes de la littérature. Ici :

- La notion d'abstraction. Un interacteur est une entité numérique pure qui, en référence au modèle d'architecture logicielle Arch [UIMS 92], couvre les niveaux présentations logique et physique. La présentation logique gère la fonction de service. La présentation physique en assure le rendu et la manipulation par l'utilisateur, sous réserve de capteurs/effecteurs et d'entités physiques compatibles pour les actions/observations humaines et système ;
- La granularité. Le grain d'un interacteur est variable : c'est celui de sa fonction de service. En effet, un interacteur est élémentaire s'il rend une fonction élémentaire.
- L'interaction. L'interacteur est un médiateur direct entre les acteurs humains et artificiels. C'est sa projection dans son milieu environnant (les entités physiques) qui définit la frontière d'échange entre l'utilisateur et le système. Selon la nature de sa fonction de

- service (si elle est typiquement d'estime), l'interacteur peut être réduit à une seule fonction de consultation ;
- La modalité. Aucun préjugé n'est ici fait quant à la nature des entités physiques, capteurs et effecteurs. Les interacteurs dépassent donc le seul cadre du graphique.

Jusqu'ici, ces descriptions (fonction de service, couplage d'entités, valeur) restaient implicites, câblées dans l'interacteur, interdisant en conséquence aux interacteurs toute transformation opportuniste. Pourtant, si l'utilisateur typiquement souhaite disposer de sa main gauche pour un autre usage, il pourrait être pertinent d'afficher le clavier du téléphone sur l'objet le plus proche de taille et de forme compatibles. Rendre explicites ces informations pour un couplage dynamique des entités, telle est la ligne de force de notre vision. Le choix entre entités candidates sera dirigé par la valeur.

La définition et les concepts de base étant posés, la section suivante dresse notre vision des interacteurs en plasticité.

## 2. Vision des interacteurs pour la plasticité

La définition fonctionnelle donnée aux interacteurs contraste avec les boîtes à outils d'aujourd'hui qui valorisent plus la présentation des widgets que leur fonction. La sémantique reste implicite, véhiculée par leur seule présentation. Les boutons radio, par exemple, n'expriment pas leur pertinence pour des tâches de sélection d'une option parmi N. Les cases à cocher ne racontent pas leur adéquation pour une sélection de M options parmi N. Le savoir-faire des concepteurs est alors primordial au risque sinon de transgresser le critère de compatibilité [Bastien 93]. Dans Word, par exemple, les cases à cocher ne sont pas appropriées pour le paramétrage des fontes où certaines options sont deux à deux exclusives (Figure 13). L'anomalie est-elle réfléchie, privilégiant volontairement l'esthétique à la compatibilité ? L'histoire ne le dit pas. Pourquoi l'IHM ne le raconterait-elle pas ?

Attributs		
<input checked="" type="checkbox"/> Barré	<input checked="" type="checkbox"/> Ombré	<input checked="" type="checkbox"/> Petites majuscules
<input type="checkbox"/> Barré double	<input checked="" type="checkbox"/> Contour	<input type="checkbox"/> Majuscules
<input type="checkbox"/> Exposant	<input type="checkbox"/> Relief	<input checked="" type="checkbox"/> Masqué
<input checked="" type="checkbox"/> Indice	<input type="checkbox"/> Empreinte	

Figure 13 : Cases à cocher inappropriées pour des options deux à deux exclusives.

Pourquoi l'IHM ne raconterait-elle pas non plus que les libellés qui annoncent des champs texte ont un rôle d'incitation (critère de guidage, sous-critère d'incitation dans le référentiel de [Bastien 93]) ? Idem pour les valeurs et formats acceptables (Figure 14) ? Ces widgets sont aujourd'hui juxtaposés, laissant leur complémentarité à un stade au mieux documentaire.



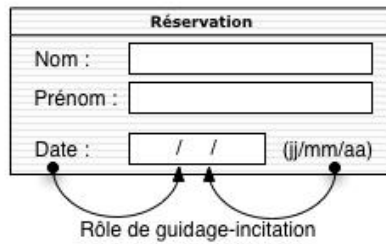


Figure 14 : L'IHM est une juxtaposition de widgets dans laquelle widgets et IHM ignorent leur raison d'être.

L'idée serait ici de :

- Recentrer les interacteurs sur leur raison d'être (c'est-à-dire leur fonction de service) et rassembler ainsi en un même interacteur les widgets dont la complémentarité offre la fonction attendue. Sur la figure 14, par exemple, on verrait trois interacteurs permettant de « spécifier le nom », « spécifier le prénom » et « spécifier la date ». Ces interacteurs seraient composés respectivement d'un libellé d'incitation et d'un champ texte pour le nom et le prénom ; d'un libellé, d'un champ texte et d'un autre libellé pour la date, ce deuxième libellé indiquant le format acceptable ;
- Encapsuler en un même interacteur les présentations fonctionnellement équivalentes et en exprimer les requis (en termes de ressources) et propriétés. Par exemple, « spécifier nom » pourrait englober les deux présentations de la Figure 15.



Figure 15 : Deux présentations fonctionnellement équivalentes. L'idée serait de les encapsuler en un même interacteur dont la raison d'être serait « spécifier nom ».

- Eventuellement, d'assortir les interacteurs de capacités d'adaptation. Ces capacités seraient alors des fonctions techniques d'adaptation.

La conception devrait non seulement en être facilitée, mais sa pérennité assurée et son adaptation préparée. C'est le fondement de notre vision. Cette vision cible, bien entendu, la plasticité, mais avec l'ambition conjointe d'améliorer la conception, l'exécution et l'évaluation d'IHM plus traditionnelles. Nous examinons ci-dessous trois points de vue : le créateur d'interacteurs, le concepteur de systèmes interactifs et l'utilisateur final de ces systèmes interactifs.

## ➡ Créateur d'interacteurs

Pour le créateur d'interacteurs, l'idée est d'élaborer et de décrire des flots de compétences correspondant aux fonctions de service de l'interacteur. La description

porte sur les fonctions (de service, techniques et contraintes) de l'interacteur et ses présentations. Par exemple, l'interacteur « *Sélecteur exclusif* » raconte qu'il rend la fonction de service « *Sélectionner 1 option parmi N* » pour laquelle il dispose de trois présentations :

- Les boutons radio, appropriés pour un choix entre cinq et six options [Nogier 05]. Ils rendent en réalité la fonction de service « *Sélectionner 1 option parmi 6 au maximum* » ;
- Les menus, pertinents jusqu'à dix ou douze choix et intéressants pour leur encombrement spatial réduit [Nogier 05]. Ils rendent la fonction de service « *Sélectionner 1 option parmi 12 au maximum* ». Ils se distinguent des boutons radio par un coût réduit en surface d'affichage (action S) ;
- Les listes, adéquates pour plus de douze choix et dès lors que ces options sont variables [Nogier 05]. Elles rendent la fonction de service « *Sélectionner 1 option parmi plus de 12, ces options étant variables* ».

Chaque présentation se décrit en termes de :

- Fonctions de service, techniques et contraintes ;
- Apparence et comportement (look and feel) ;
- Ressources physiques, numériques et cognitives d'une part requises, d'autre part consommées pour les actions et observations humaines et système. Ces descriptions seront fondamentales pour évaluer la viabilité d'une présentation dans un contexte d'usage donné, mesurer sa valeur et envisager, si nécessaire, une adaptation. Autant le coût physique sera absolu, autant le coût cognitif sera (en toute prudence) relatif, dépendant de l'utilisateur ;
- Propriétés assurées. Ces propriétés pourront être calculées par le système (par exemple, la longueur de la trajectoire d'interaction) ou affirmées par le concepteur (par exemple, l'affordance d'une présentation).

L'adaptation peut être câblée dans l'interacteur ou spécifiable par le concepteur et/ou l'utilisateur final. L'important est que l'interacteur décrive ses fonctions techniques d'adaptation pour qu'un raisonnement puisse, si nécessaire, être tenu sur l'adaptation globale du système interactif. La description comprend l'expression :

- Du changement de contexte d'usage justifiant l'adaptation ;
- Des ressources requises et consommées par l'adaptation ;
- Ainsi que des éventuelles propriétés garanties, par exemple, la continuité de l'interaction [Trevisan 04] [Denis 04].

Nous regrouperons sous le terme de *qualité de service* les volets ressources et propriétés. La figure 16 résume les descriptions ainsi attendues pour un interacteur.

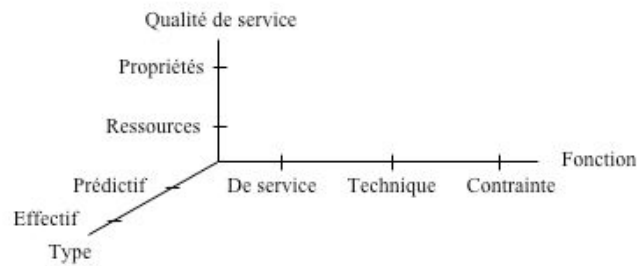


Figure 16 : Cahier des charges pour la description des interacteurs en plasticité.

La description des fonctions et de la qualité d’usage peuvent être prédictives à la conception ou effectives à l’exécution.

L’ambition est d’offrir au concepteur un environnement de prototypage rapide à base d’interacteurs façonnés pour la plasticité. Le concepteur choisirait ses interacteurs dans une palette d’outils et parcourrait alors, via par exemple une molette (Figure 17), les présentations disponibles. Il consulterait leurs requis et propriétés puis les brancherait, si pertinents, au système interactif.

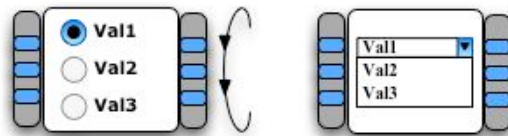


Figure 17 : Vers un environnement de prototypage rapide dans lequel le concepteur parcourrait, via par exemple une molette, les présentations disponibles pour un interacteur donné (ici le sélecteur exclusif).

### ☛ Concepteur de systèmes interactifs

Pour le concepteur de systèmes interactifs, l’idée est de mettre en correspondance les besoins utilisateur et les fonctions de service offertes par les interacteurs. Typiquement, dans une approche descendante à base de modèles des tâches, le concepteur affinera son modèle des tâches jusqu’à un niveau dit “tâches élémentaires” [Balbo 94] sur lequel il « branchera » les interacteurs, gardant ainsi trace du rôle de chaque interacteur dans le système interactif global. Le concepteur spécifiera alors l’adaptation du système interactif selon le contexte d’usage, si cette adaptation n’est toutefois pas déjà embarquée dans l’interacteur.

## Utilisateur final

Pour l'utilisateur final, une retombée de la rationalisation de la conception et de son existence à l'exécution est l'auto-explication des IHM. Les IHM racontent leur raison d'être. On imagine que, par des zooms progressifs, l'utilisateur "voyage" au cœur de l'IHM, dépassant la seule façade pour en comprendre la logique d'utilisation. L'interacteur se décrit intrinsèquement et extrinsèquement racontant, en particulier, la tâche utilisateur qu'il remplit et les relations de voisinage qu'il entretient. Sur la Figure 18, par exemple, la liste de gauche explique qu'elle permet la sélection d'une œuvre, ensuite explorée en partie droite. Une lentille magique [Bier 93] est ici imaginée pour accéder à la description intrinsèque de l'interacteur : la liste raconte qu'elle rend observable un ensemble de concepts affichables sous la forme d'un texte (c'est sa fonction de service). Cet ensemble de concepts est ici une galerie de peintures.

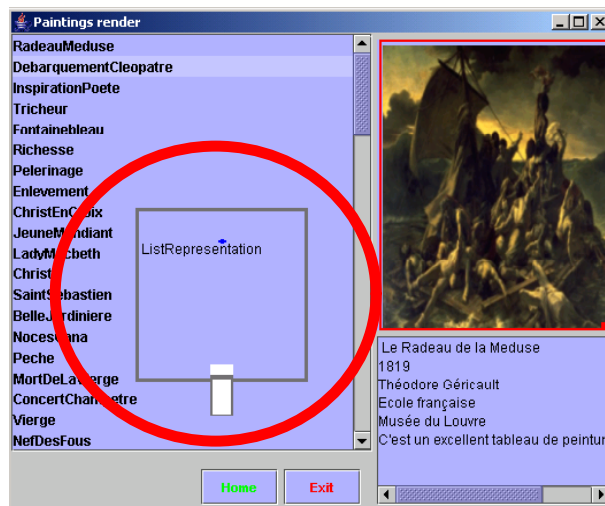


Figure 18 : Une lentille magique permet d'explorer les dimensions jusqu'ici cachées de l'interacteur. Elle rend ici observable sa fonction de service : le choix d'une œuvre dans une galerie de peintures. Extrait du démonstrateur DECOR (Chapitre 5).

Dans cette vision, la difficulté revient au créateur d'interacteurs. Aussi, est-il important de cerner les requis et propriétés des interacteurs. C'est l'objet de la section suivante.

## 3. Exigences pour la plasticité

La vision appelle comme requis l'autodescription de l'interacteur et une éventuelle prédisposition à l'adaptation par un potentiel de présentations variées et de mécanismes d'adaptation. Nous identifions donc deux exigences : une capacité d'autodescription et un potentiel d'adaptation.

## ☛ Autodescription

Nous nous appuyons sur la Théorie du Système Général [Le Moigne 77] pour identifier trois volets dans la description : une description fonctionnelle, ontologique et génétique. Nous ajoutons un quatrième volet pour la description de la valeur.

- La description fonctionnelle porte sur les fonctions métier de l'interacteur et ses présentations. Considérons le système interactif PlasticClock [Calvary 04] (Figure 19). PlasticClock est un démonstrateur de plasticité. Il permet à un usager de consulter l'heure dans deux villes et éventuellement la date selon la place disponible à l'écran. Lorsque l'utilisateur étire la fenêtre en hauteur alors la présentation des heures passe d'un format texte (a) à un format cadran (b) puis double cadran (c). La date est ajoutée lorsque l'utilisateur élargit la fenêtre (d). Dans ce volet fonctionnel, PlasticClock doit raconter sa fonction métier de consultation de l'heure et éventuellement la date dans deux villes. Trois présentations sont possibles : texte, cadran et double cadran.
- La description ontologique est relative à l'objet en tant qu'être. Elle couvre son état, sa structure et son interface en tant qu'API (Application Programming Interface). L'état couvre à la fois l'abstraction et la présentation de l'interacteur. La connaissance de cet état est fondamentale pour, en particulier, une reprise de l'interaction à plus fin grain que la session. Typiquement, si l'utilisateur avait déjà spécifié la date du jour, il apprécierait post-adaptation de retrouver cette date dans l'IHM adaptée. Cette préservation de l'état aura un impact fort sur la charge de travail telle qu'exprimée dans le référentiel de C. Bastien et D. Scapin [Bastien 93]. La structure est relative à la structure interne de l'interacteur. Dans l'hypothèse où PlasticClock serait elle-même un interacteur, la structure de PlasticClock décrirait pour la quatrième présentation (Figure 19d) l'assemblage de deux interacteurs : « Consulter l'heure dans deux villes » et « Consulter date ». L'API décrit le contrat syntaxique de l'interacteur. Ce contrat est vital pour la connexion de l'interacteur à d'autres entités numériques ;
- La description génétique ajoute à l'Être (ontologique) et au Faire (fonctionnelle) le Devenir conformément aux travaux de Piaget : « Définir l'objet, c'est le connaître dans son histoire (hérédité) et donc son projet (son devenir) ». Nous distinguons ici le passé de l'interacteur (son historique) de son devenir. L'historique peut être intéressant pour, par exemple, apprendre les préférences utilisateur et ajuster en conséquence l'adaptation de l'interacteur. Si typiquement, l'utilisateur rétrécit toujours la fenêtre de PlasticClock pour n'en voir que l'heure, alors l'affichage de la date (Figure 19d) n'est peut-être pas pertinent. Le devenir est relatif à l'adaptation ;

- La description de la valeur se fait selon les deux aspects de l'économie [Chevallier 89], sans par contre qu'un jugement de valeur ne soit fait : nous ne considérons pas le ratio « satisfaction du besoin / coût » mais ces deux aspects. La satisfaction du besoin est liée au rôle que tient l'interacteur dans le système interactif (par exemple la tâche utilisateur qu'il réalise) et aux propriétés qu'il garantit. Le coût d'exploitation mesure la consommation effective des ressources physiques, numériques et cognitives pour les parties métier et adaptation, abstraction et présentation de l'interacteur.

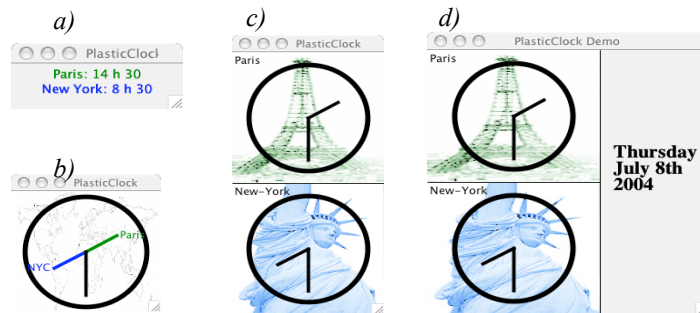


Figure 19 : *PlasticClock, un démonstrateur de plasticité. L'IHM se remodèle selon la place disponible à l'écran.*

Illustrons ces dimensions sur PlasticClock. Dans l'hypothèse où un interacteur « Consulter l'heure dans deux villes » serait disponible, l'autodescription serait la suivante (Figure 20) :

<u>Autodescription</u>	
- Fonctionnelle :	<ul style="list-style-type: none"> <li>- Fonction métier : Consulter l'heure dans deux villes</li> <li>- Présentations : <ul style="list-style-type: none"> <li>o 1) Figure 19a : <ul style="list-style-type: none"> <li>« Ville1 : » heure1 « h » minutes1</li> <li>« Ville2 : » heure2 « h » minutes2</li> </ul> </li> <li>o 2) Figure 19b : cadran à 3 aiguilles</li> <li>o 3) Figure 19c : double cadran</li> </ul> </li> </ul>
- Ontologique :	<ul style="list-style-type: none"> <li>- Etat (au lancement): <ul style="list-style-type: none"> <li>o Abstraction : Paris, 14h30 ; New-York, 8h30</li> <li>o Présentation : Figure 19a</li> </ul> </li> <li>- Structure : agent PAC</li> <li>- API : non décrite ici</li> </ul>
- Génétique :	<ul style="list-style-type: none"> <li>- Historique : Figure 19a au lancement</li> <li>- Adaptation : notons <math>tf</math> la taille de la fenêtre. Si (sortie( <math>tf \in [xi-1, yi-1]</math> ) et entrée ( <math>tf \in [xi, yi]</math> )) Alors commuterDe_A_(présentationi-1, présentationi)</li> </ul>
- Valeur pour les 3 présentations :	<ul style="list-style-type: none"> <li>- Ressources physiques : celles liées au graphique <ul style="list-style-type: none"> <li>o Action (S) : projecteur, surface d'affichage.<math>tf(i)</math>, <math>i \in [1 .. 3]</math> (n° présentation)</li> <li>o Observation (H) : yeux, surface d'affichage</li> <li>o Action (H) : -</li> <li>o Perception (S) : -</li> </ul> </li> <li>- Propriétés : compatibilité avec les standards tel que ISO</li> </ul>

Figure 20 : Autodescription de l'interacteur " Consulter l'heure dans deux villes".

## ➤ Potentiel d'adaptation

Nous distinguons deux volets dans le potentiel d'adaptation :

- Des ressources en termes d'abstractions et de présentations. Elles confèrent alors à l'interacteur un caractère polymorphe ;
- Des mécanismes pour la mise en œuvre de l'adaptation. Ces mécanismes sont relatifs à l'adaptation (détection du changement de contexte d'usage, calcul de réaction, mise en œuvre de la réaction [Calvary 01]) et au contrôle de l'exécution de l'interacteur (capture et restauration d'état ; démarrage, arrêt, suspension et reprise [De Palma 99]). Le calcul de la réaction peut se limiter à la considération des présentations préfabriquées. Mais il peut aussi supposer l'existence d'outils de génération construisant à la volée les présentations souhaitées. Si tous ces mécanismes (d'adaptation et de contrôle d'exécution) ne sont pas internes à l'interacteur, ils doivent être assurés par une infrastructure d'exécution externe [Oreizy 99]. Ils font donc partie du coût de l'interacteur en tant que ressources numériques requises.

La Figure 21 regroupe en une grille d'analyse les exigences qui ont ainsi émergé pour la construction et/ou la critique d'interacteurs dans une perspective de plasticité.

Approches	Autodescription									Potentiel d'adaptation						
	Fonctionnelle		Ontologique			Génétique		Valeur		Polymorphisme		Mécanismes d'adaptation			Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution

Figure 21 : Grille d'analyse des interacteurs pour la plasticité des IHM.

Dans cette grille, nous notons par :

- Une croix (x) les aspects de l'autodescription couverts par l'interacteur ;
- Un I / E la capacité de l'interacteur à assurer en interne (I) versus en externe (E) l'adaptation ;
- Un fond coloré la préoccupation dominante des approches. Les cases seront blanches par défaut.

Des points d'interrogation seront marqués en cas de doute.

La section suivante caractérise sur cette grille le profil minimal souhaité pour un interacteur en vue de la plasticité.

#### 4. Profil minimal requis

Bien entendu, un profil où toutes les cases seraient cochées serait optimal pour la plasticité. Au minimum, nous demandons une autodescription complète de l'interacteur.

Approches	Autodescription									Potentiel d'adaptation						
	Fonctionnelle		Ontologique			Génétique		Valeur		Polymorphisme		Mécanismes d'adaptation			Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
	x	x	x	x	x	x	x	x	x							

Figure 22 : Profil minimal requis.



Dans ce chapitre nous avons identifié les requis pour la plasticité. Ces requis portent sur deux niveaux : l'autodescription et le potentiel d'adaptation. Seule l'autodescription est requise. Le chapitre suivant examine l'état de l'art en matière d'interacteurs. Il caractérise chaque approche sur cette grille d'analyse.

## CHAPITRE II

### ETAT DE L'ART DES INTERACTEURS POUR LA PLASTICITE : REVUE ET CARACTERISATION

Les interacteurs sont un élément clé dans la construction d'IHM. Aussi, les travaux sont-ils nombreux sur ce thème. Nous n'établissons pas ici un état de l'art général des interacteurs : l'état de l'art est dirigé par les exigences identifiées au chapitre précédent. L'objectif est double :

- Identifier dans la littérature des travaux pertinents en regard de nos exigences ;
- Montrer en corollaire l'originalité et la force de notre approche.

Nous nous servons de la grille d'analyse établie au chapitre précédent pour structurer l'état de l'art. Cette grille d'analyse fait émerger, dans la partie autodescription, quatre volets : la description fonctionnelle, ontologique, génétique et de valeur de l'interacteur. A notre connaissance, aucune approche ne s'est jusqu'ici focalisée sur la valeur même si elle sous-tend parfois l'argumentaire. En conséquence, nous structurons l'état de l'art en trois sections selon la prédominance fonctionnelle, ontologique ou génétique des travaux :

- Dans le volet fonctionnel, nous plaçons les études traitant d'un éventuel polymorphisme de l'interacteur en fonction comme en présentation (Figure 23) ;

Approches	Autodescription									Potentiel d'adaptation						
	Fonctionnelle		Ontologique			Génétique		Valeur		Polymorphisme		Mécanismes d'adaptation		Contrôle d'exécution		
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution




Figure 23 : Prise en compte, dans le volet fonctionnel, des approches traitant du polymorphisme.

- Dans le volet ontologique (partie Etat), nous intégrons les travaux traitant du contrôle d'exécution de l'interacteur (Figure 24) ;

Approches	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétique		Valeur		Polymorphisme		Mécanismes d'adaptation			Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution

Figure 24 : Prise en compte, dans le volet ontologique (partie Etat), des travaux traitant du contrôle d'exécution.

- Dans le volet génétique (partie Adaptation), nous intégrons les approches traitant des mécanismes d'adaptation (Figure 25).

Approches	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétique		Valeur		Polymorphisme		Mécanismes d'adaptation			Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution

Figure 25 : Prise en compte, dans le volet Génétique (partie Adaptation) des travaux traitant des mécanismes d'adaptation.

Les trois sections (prédominances fonctionnelle, ontologique et génétique) sont traitées dans cet ordre. Nous terminons par une prise de recul sur l'existant.

## 1. Prédominance fonctionnelle

Nous répertorions ici les travaux à dominance fonctionnelle. Nous les nuancions selon qu'ils se focalisent sur la fonction de l'interacteur, sa présentation ou son polymorphisme.

## 1.1 Fonction métier

Les interacteurs ont toujours été implicitement définis par rapport à la tâche utilisateur qu'ils supportent. Son explicitation fut un moyen pour raisonner sur d'autres préoccupations. Par exemple :

- Au grain des systèmes interactifs, la génération d'IHM multicibles (une cible étant définie par un contexte d'usage) ;
- Au grain des interacteurs, le polymorphisme ou l'auto-adaptation.

Dans ArtStudio, D. Thevenin [Thevenin 01] typiquement traite de la génération d'IHM multicibles. Il définit un interacteur comme «un système capable de représenter une information auprès de l'utilisateur et lui offrir éventuellement des tâches de manipulation de cette information ». D. Thevenin se limite aux interacteurs graphiques. Il en propose la modélisation suivante (Figure 26) :

- « Un interacteur manipule une donnée. Cette donnée correspond à l'interface d'un flux bidirectionnel (entrée et sortie) ;
- Il a obligatoirement une représentation graphique – nous imposons un retour d'information (feedback) graphique à l'utilisateur ;
- Il permet la réalisation d'une tâche. Selon la tâche, nous savons si l'interacteur est d'entrée (ex : sélectionner) ou de sortie (ex : consulter) » [Thevenin 01].

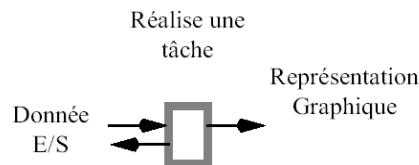


Figure 26 : Modèle des interacteurs de ArtStudio [Thevenin 01].

Même si notre définition des interacteurs ne correspond pas tout à fait à celle de D. Thevenin (ouverture à des flux non bidirectionnels et à des modalités autres que le graphique), nous retenons de ces travaux :

- L'autodescription en termes de tâches, de concepts du domaine (données E/S) et de présentation. La description en termes de tâches s'appuie sur une taxonomie de tâches élémentaires. La description des concepts du domaine pointe vers un modèle dédié. La présentation s'appuie sur les widgets d'utilité publique dont les hauteur et largeur sont ici spécifiés. Ces requis sont confrontés à un modèle de plate-forme lors de la génération semi-automatique de l'IHM. La Figure 27 donne un exemple de spécification d'interacteur dans le modèle d'ArtStudio [Thevenin 01]. Cette spécification comprend des informations telles que les valeurs de présentation (hauteur et largeur) de l'interacteur, les types de données qu'il manipule ou aussi la tâche qu'il supporte ;

```

<GraficInteractor>
<name>XComboBox</name>
<width value="null"/>
<height value="20"/>
<data list = "{TString,TStringFromTVector}"/>
<elementaryTask list = "{selectionner}"/>
<type value="visualisation"/>
</GraficInteractor>

```

Figure 27 : Spécification d'un interacteur dans ArtStudio [Thevenin 01].

- La taxonomie de tâches élémentaires. D. Thevenin définit une tâche élémentaire comme étant « la plus petite tâche d'interaction indépendante de la plate-forme. Si elle devait être décomposée plus avant, on obtiendrait des tâches d'interaction dépendantes de la cible ». Cette définition est motivée par la préoccupation de D. Thevenin : la génération d'IHM multicibles. Elle suppose que toute tâche utilisateur se décompose récursivement en un ensemble de sous-tâches, jusqu'à obtenir des tâches élémentaires [Balbo 94]. Dans sa taxonomie, D. Thevenin distingue les tâches d'utilité publique (ex : sélectionner une option parmi N) des tâches métier propres à un domaine applicatif donné (ex : admirer une galerie de peintures). Les tâches d'utilité publique sont celles offertes par les boîtes à outils usuelles. Dans la lignée de Foley [Myers 90], D. Thevenin en propose une classification : il identifie les tâches de sélection, spécification, activation et consultation.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phisme		Mécanismes d'adaptation		Contrôle d'exécutio	
Approches	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Génére	Détection delta cxt	Calcul de réaction	Mise en oeuvre	Gestion d'état	Gestion d'exécution
ArtStudio	x	z						z								

Figure 28 : Caractérisation des interacteurs de ArtStudio [Thevenin 01].

Tandis que D. Thevenin s'intéresse à une description générale des interacteurs (Figure 28), d'autres auteurs s'attachent à en inventer de nouveaux. Par exemple, les Toolglass de Bier [Bier 93]. Une toolglass est une surface transparente, sur laquelle sont disposés des outils (interacteurs) semi-transparents. La Figure 29 montre l'exemple d'une palette de couleurs. Il suffit de placer la couleur désirée sur l'objet à peindre puis de cliquer sur cette couleur pour changer la couleur de l'objet situé en dessous. Un tel bouton est appelé « Click through button ». D'autres fonctions (création, suppression, etc.) sont, bien entendu, envisageables.

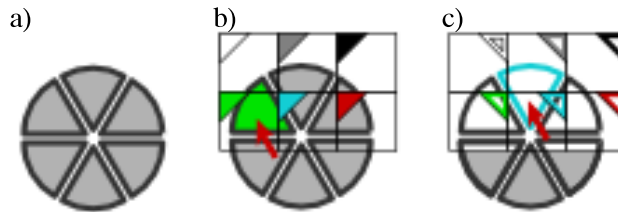


Figure 29 : Des toolglass pour modifier la couleur des objets [Bier 93].

Dans la description de cet interacteur, il serait intéressant de rendre explicites les différences par rapport au bouton classique, à savoir :

- Le principe de couplage entre, d'une part, l'objet à peindre et, d'autre part, le « Click through button » : c'est de la superposition de ces deux objets qu'émerge la fonction « modifier couleur » ;
- Une consommation de surface d'affichage partagée entre l'objet à peindre et le « Click through button » : la superposition des deux objets, encouragée par la semi-transparence du bouton, donne lieu à une économie de surface d'affichage.

Une application originale en plasticité serait l'inspection d'un interacteur par un « Click through button ». En cas de polymorphisme, l'interacteur exposerait alors ses différentes « formes » donnant à l'utilisateur le loisir de choisir celle qui lui semble la plus appropriée. La fonction du « Click through button » relèverait alors de la méta-IHM, c'est-à-dire de l'IHM de l'adaptation [Roudaut 06].

M. Florins [Florins 06] s'appuie sur la modélisation de D. Thevenin pour la transformation d'IHM lors d'une dégradation de plate-forme. Elle en développe la partie présentation.

## 1.2 Présentation

M. Florins [Florins 06] traite de la dégradation élégante d'IHM. Le terme de *dégradation* ne porte pas sur l'IHM (qui sera peut-être augmentée dans la transformation) mais sur la plate-forme d'exécution : elle propose une méthode de transformation d'IHM pour passer d'une version "luxe" (grand écran ou fortes capacités, etc.) à une version plus réduite (plus petit écran ou faible mémoire, etc.).

L'incrément par rapport à D. Thevenin se situe dans la modélisation de la présentation. M. Florins s'appuie sur la "colonne vertébrale" du cadre de référence CAMELEON (Figure 30) pour identifier trois niveaux d'abstraction dans la modélisation. Elle s'inscrit ainsi dans la lignée des travaux de J. Vanderdonck sur les OIA-OIC [Vanderdonck 97] (Objet d'Interaction respectivement Abstrait et Concret) sans toutefois que les niveaux d'abstraction se correspondent parfaitement. M. Florins adopte trois niveaux de description :

- Une description abstraite en termes d'interacteurs abstraits étendus (extendedAIO). Par exemple : un spécifieur d'entier, un sélecteur de date, un séparateur ou un conteneur. Cette description relève du niveau Tâches-Concepts en référence à CAMELEON. Elle n'a pas de correspondant en termes d'OIA ou d'OIC ;
- Une description concrète en termes d'interacteurs "logiques", c'est-à-dire conceptuels par opposition à une préoccupation "physique" qui serait implémentationnelle, c'est-à-dire relative à la mise en œuvre de ces interacteurs dans une boîte à outils particulière (par exemple, Swing). Les champs texte, boutons radio, cases à cocher, etc. relèvent de ce niveau d'abstraction. Plusieurs boîtes à outils les implémentent (par exemple, Motif, Swing, etc.). Dans une approche descendante (forward engineering), cette description est une réification du niveau abstrait (Tâches-Concepts). Elle correspond au niveau Interface concrète de CAMELEON. Son correspondant chez J. Vanderdonck est l'OIA. Elle pointe vers un modèle de plate-forme sur lequel elle exprime des requis ainsi que l'utilisabilité qu'elle garantit : les deux volets de la valeur (satisfaction du besoin et coût) y sont donc intégrés. Mais ils ne sont pas modélisés. L'hypothèse est que l'interacteur concret (graphicalCIO) sait calculer et retourner son utilisabilité sur une plate-forme donnée (c.f. la méthode usabilityOnPlatform()) ;
- Une description finale correspondant à l'interacteur tel qu'offert dans les boîtes à outils. C'est une réification du niveau concret (graphicalCIO) en un finalWidget. Chez J. Vanderdonck, le correspondant est l'OIC.

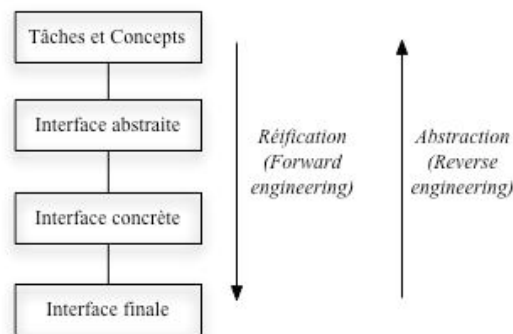


Figure 30 : Colonne vertébrale du cadre de référence CAMELEON.

Il est intéressant de noter dans ce travail que :

- Les interacteurs sont limités au graphique : M. Florins cible les IHM graphiques ;

- Le niveau interface abstraite de CAMELEON n'est pas considéré : les interacteurs y sont supposés atomiques, c'est-à-dire non décomposables.

Notre définition des interacteurs va au-delà de ces hypothèses. Néanmoins, nous retenons :

- Le principe de la modélisation multifacettes : un interacteur peut supporter une ou plusieurs tâches. Il comportera autant de facettes qu'il offrira de tâches ;
- Le polymorphisme à tout niveau d'abstraction : dans la lignée des OIA-OIC où plusieurs OIC pouvaient correspondre à un même OIA, l'extendedAIO de M. Florins peut être réifié en plusieurs graphicalCIO, eux-même réifiables en plusieurs finalWidgets. En revanche, chez M. Florins, la réification est obligatoire : M. Florins n'envisage pas qu'un extendedAIO et graphicalCIO ne soient pas réifiés ;
- Le principe d'une hiérarchie de tâches (Figure 31) et de concepts. Les hiérarchies sont intéressantes pour soutenir les transformations. Une stratégie peut être, par exemple, de préserver la tâche utilisateur (ex : sélectionner une option parmi N) et de ne la dégrader (ex : spécifier) qu'en cas d'impasse (pas de finalWidget disponible pour la plate-forme cible).

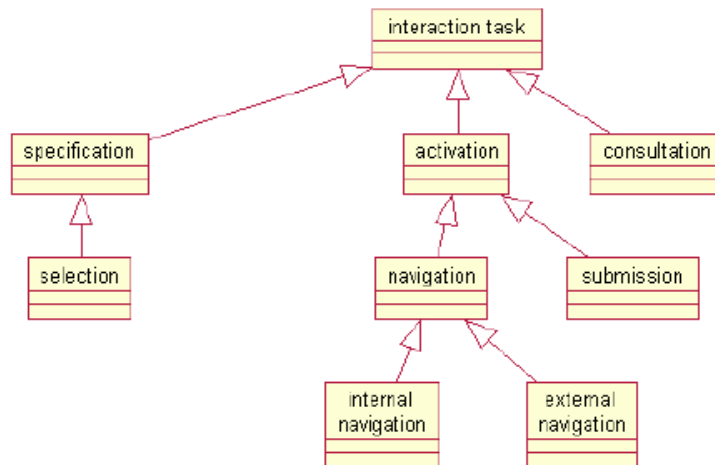


Figure 31 : Hiérarchie de types de tâches dans M. Florins. Extrait de [Florins 06].

La Figure 32 organise les concepts impliqués dans le modèle d'interacteur de M. Florins. Elle s'appuie essentiellement sur Teallach [Griffiths 01] et UsiXML (<http://www.usixml.org>) pour le nommage des *extendedAIO* et *graphicalCIO*.



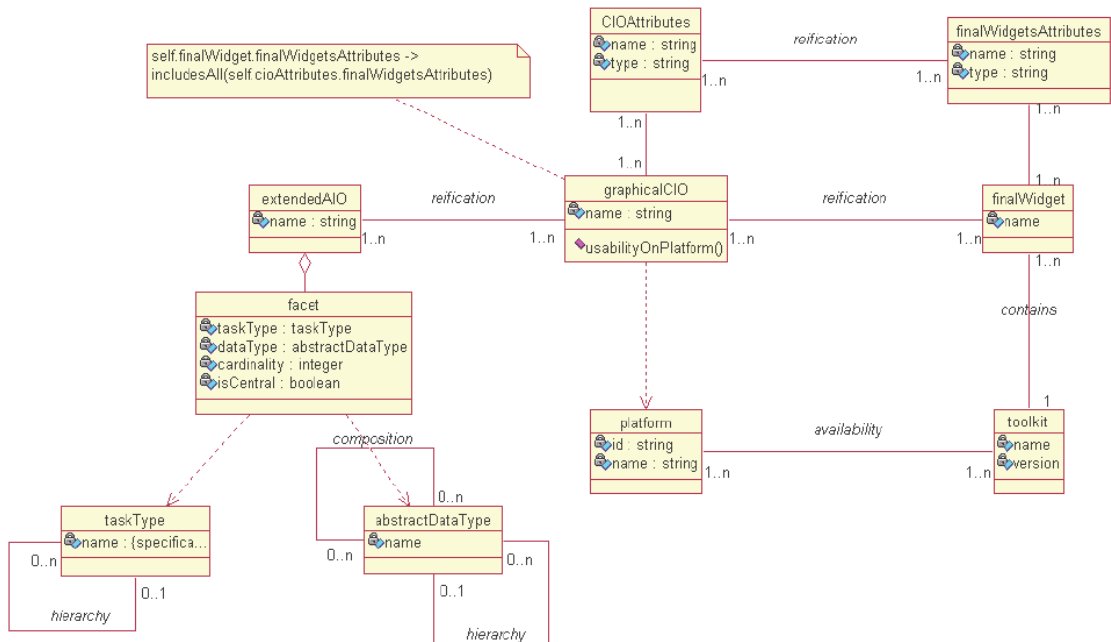


Figure 32 : Concepts au cœur du modèle d'interacteur de M. Florins. Extrait de [Florins 06].

La Figure 33 caractérise le modèle de M. Florins dans notre grille d'analyse. La préoccupation dominante du modèle est le polymorphisme au niveau des présentations.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phisme		Mécanism es d'adaptation		Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection de la cxtx	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
Approches																
M. Florins	x	x						~	~							

Figure 33 : Caractérisation du modèle de M. Florins [Florins 06].

Si M. Florins se limite au graphique, d'autres travaux explorent de nouvelles modalités ou alliances de modalités. Pour exemples :

- Les widgets multimodaux de S. Brewster [Brewster 97] (les Sonically enhanced widgets) qui allient son et graphique Le son est construit sur la base de messages audio structurés appelés Earcons [Blattner 89]. Un son est joué lorsque l'utilisateur survole ou active un widget. En termes des propriétés CARE, il y a donc redondance entre le graphique et le son pour véhiculer le critère de guidage dans ses sous-critères incitation (survole le widget) et retour d'information

(active le widget) tels qu'exprimés dans le référentiel de C. Bastien et D. Bastien [Bastien 93] ;

- Les Speech widgets de Olsen [Olsen 01] qui, de même, allient son et graphique de manière redondante. Ici, le son ne se limite, par contre, pas à des bips sonores : un message est prononcé par le widget lorsqu'il est actif. La Figure 34 en donne un exemple. Le libellé graphique du widget est "Sleep temp". Le texte prononcé est "Sleep temperature".

```
<name text="Sleep temp."
      spoken="sleep temperature"/>
or
<name text="ft." spoken="feet"/>
```

Figure 34 : Un speech widget spécifie sa présentation graphique (text) et sonore (spoken).

- Les Light widgets [Fails 02] ouvrent l'interaction homme-machine à n'importe quelle surface. La Figure 35, par exemple, dissémine de façon originale le contrôle de la télévision dans l'environnement : le bouton marche-arrêt devient l'angle haut droit de la télévision (TV Power) ; le contrôle du volume et le changement de chaînes se font sur la tête de lit (respectivement le montant gauche (Volume Adjustor) et l'arête supérieure (Station Selector)). Un Light widget est activé par tout objet dont la couleur avoisine celle de la peau. Il suffit donc à l'utilisateur de toucher la télévision ou sa tête de lit dans les zones indiquées pour soit allumer/éteindre l'appareil, soit en régler le volume/les chaînes.

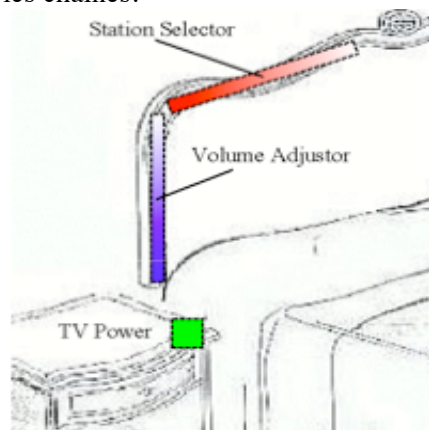


Figure 35 : Trois Light widgets pour le contrôle de la télévision (l'allumer ou l'éteindre, en régler le volume ou les chaînes).

- Les Phidgets [Greenberg 01] vont un pas plus loin que les Light widgets dans l'intégration du monde physique : alors que les Light widgets étaient limités à une interaction en entrée, les Phidgets peuvent couvrir entrée et/ou sortie. La figure 36 donne un exemple en sortie : l'utilisateur, par son doigt (objet physique), active une case

à cocher numérique projetée sur le mur physique. L'activation de cette case a pour effet d'allumer la lampe (objet physique).



Figure 36 : Phidget de manipulation d'une lampe : l'utilisateur agit sur une IHM numérique (la fenêtre) pour piloter le monde physique (allumer la lampe).

L'ouverture sur le monde physique est une tendance forte (c.f. les dates des travaux) correspondant au déconfinement des IHM. Des boîtes à outils physiques sont même proposées. Pour exemple, iStuff [Ballagas 03] qui fournit des composants physiques liés à une infrastructure logicielle. La Figure 37 en montre des exemples.

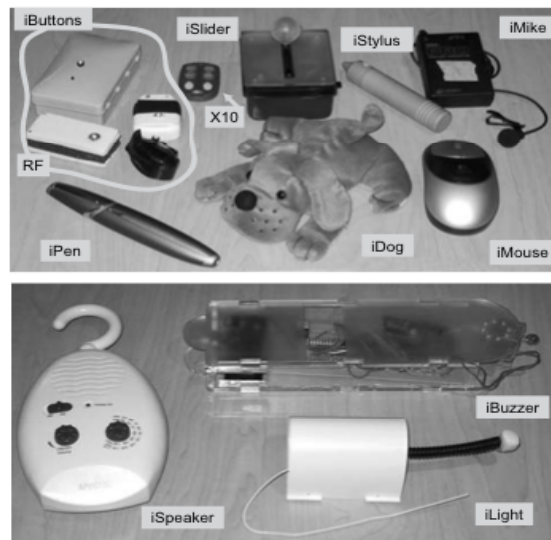


Figure 37 : Des exemples de composants iStuff pour la construction d'IHM physiques [Ballagas 03].

Dans ce même registre, mentionnons TAC (Token And Constraints) [Calvillo-Gamez 03] qui se focalise sur l'encapsulation et la composition d'objets physiques. La Figure 38 montre l'exemple de deux objets (une balle et une boîte) qui prennent alternativement les rôles de *token* et de *contrainte*. A gauche, la balle est un token dont la boîte contraint le déplacement. A droite, la boîte est un token dont la balle contraint le redimensionnement. Cet exemple est intéressant pour confirmer l'émergence et la force de la notion de *rôle* en ingénierie de l'interaction homme-machine.

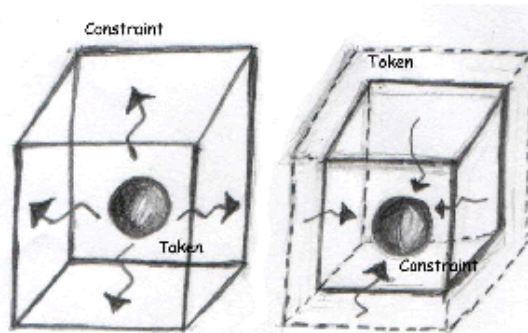


Figure 38 : *Token et Contrainte, deux rôles ici alternativement tenus par la balle et la boîte. A gauche, la boîte (contrainte) contraint le déplacement de la balle (token). A droite, la balle (contrainte) contraint le redimensionnement de la boîte (token).*

Cette préoccupation d'encapsulation et de composition est partagée par les approches à dominance ontologique, telle qu'ADC [Markopoulos 97] par exemple.

## 2. Prédominance ontologique

Les premiers travaux sur les interacteurs traitaient des préoccupations état et structure. Il est intéressant de noter que ces préoccupations font l'objet de travaux généraux en génie logiciel. De la même façon, la description de l'API est à l'étude : les travaux sur les composants, en particulier [Beugnard 99], introduisent la notion de *contrat* qu'ils déclinent en contrats syntaxiques, sémantiques, etc. selon la préoccupation ciblée. Le contrat syntaxique correspond à la description de l'API. Nous n'examinons pas ces travaux ici, ce chapitre étant borné à l'étude des interacteurs.

## 2.1 Etat

Le modèle York décrit un interacteur par un *état*, les *événements* qui manipulent cet état et les moyens par lesquels l'état est rendu perceptible et manipulable à par l'utilisateur [Duke 93b]. La description rend saillants les flux de données (Figure 39) :

- Le système est à l'origine d'événements, susceptibles de modifier l'état de l'interacteur (❶) ;
- Cet état est rendu perceptible à l'utilisateur par le biais d'une présentation (❷) ;
- La manipulation de cette présentation par l'utilisateur peut modifier l'état de l'interacteur (❸) ;
- La modification de l'état peut engendrer un événement système (❹).

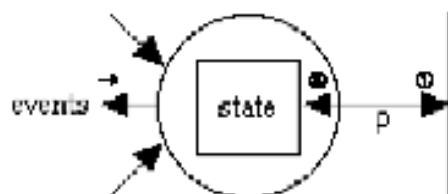


Figure 39 : Au cœur du modèle York, se trouve l'état de l'interacteur. Cet état est contrôlé par le système et l'utilisateur par le biais d'événements. Extrait de [Duke 93b].

Le modèle York met l'accent sur l'état d'un interacteur (Figure 40).

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique		Génétiq ue		Valeur		Polymor phisme		Mécanismes d'adaptation		Contrôle d'exécutio			
Approches	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Génére	Détection de la ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
York																

Figure 40 : Caractérisation du modèle d'York.

La section suivante examine le volet structure.

## 2.2 Structure

La structure relève de l'architecture logicielle. Des modèles généraux tels que MVC [Reenskaug 79] ou PAC [Coutaz 87] sont ici pertinents, mais nous nous limitons aux travaux traitant spécifiquement des interacteurs. Historiquement, le modèle Pisa [Faconti 93] décrit l'interacteur comme une boîte "assurant la communication bidirectionnelle entre l'utilisateur et le système". C'est la "conceptualisation d'un composant logiciel". G. Faconti en propose deux versions : une version boîte noire

qui exprime ce rôle d'intermédiaire entre l'utilisateur et le système ; une version boîte blanche qui décrit la structure interne de l'interacteur (Figure 41). Cette structure s'appuie sur quatre processus (collection, feedback, control et mesure) alimentés par des événements externes (receive et trigger) susceptibles de générer eux-mêmes des événements à portée interne ou externe (output). La *collection* maintient la représentation abstraite du rendu de l'interacteur. Le *feedback* gère les primitives graphiques de rendu. La *mesure* accumule et interprète les entrées de l'utilisateur. Le *control* livre à l'application ou d'autres interacteurs l'information établie en interne. Les échanges d'information avec l'environnement (utilisateur et/ou système) s'effectuent via des portes. Les portes *ir* et *or* permettent à l'interacteur de recevoir des données en provenance respectivement de l'utilisateur et de l'application. Les portes *is* et *os* assurent la transmission des données à respectivement l'application et l'utilisateur. Les deux déclencheurs *ot* et *it* permettent un contrôle synchrone de l'interacteur.

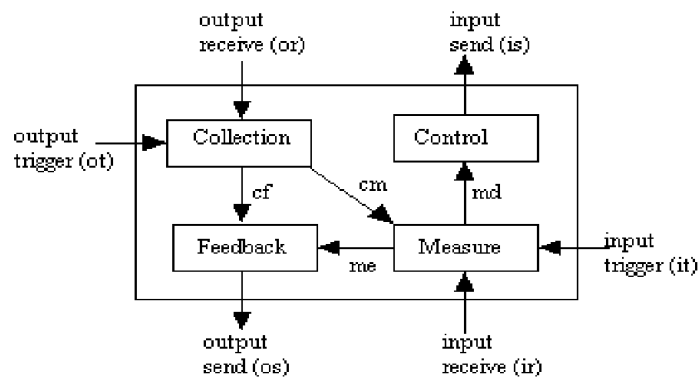


Figure 41 : Dans le modèle Pisa, l'interacteur est décomposé en quatre processus (collection, feedback, control et mesure) en interaction avec l'environnement (utilisateur et système). Extrait de [Faconti 93].

La Figure 42 caractérise le modèle Pisa sur notre grille d'analyse. Pisa est un modèle qui met l'accent sur l'aspect structure de l'interacteur.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq e		Valeur		Polymor phis me		Mécanismes d'adaptation		Contrôle d'exécutio		
Approches	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Génére	Détection delta cxtxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
Pisa																

Figure 42 : Caractérisation du modèle Pisa.

Dans cette même lignée, ADC (Abstract-Display-Control) [Markopoulos 97] définit les interacteurs comme étant " des abstractions primitives utilisées dans la description d'un système interactif. Ces abstractions peuvent être considérées comme des abstractions d'architecture logicielle similaires aux objets dans la programmation orientée objet". Les interacteurs ADC se connectent entre eux via la notion de *porte*. Une porte est caractérisée par un sens (entrée ou sortie) et un rôle qui exprime la nature de l'interaction. La Figure 43 donne la structure interne d'un interacteur ADC ainsi que les portes qui assurent les échanges entre l'interacteur et son environnement. Nous ne détaillons pas les portes. En revanche, nous notons que les trois fonctions qui émergent dans l'architecture (Abstraction, Display et Controller) coïncident avec les fonctions du modèle PAC (Présentation pour Display, Abstraction, Control pour Controller). Restent à vérifier les principes de communication mais ce n'est pas l'objet de l'étude ici.

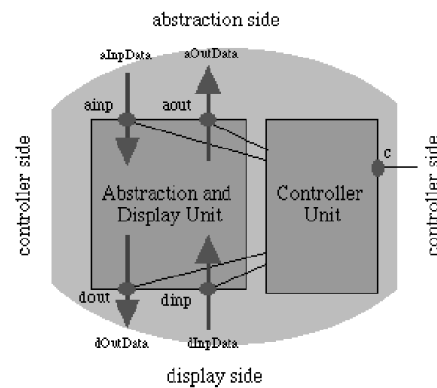


Figure 43 : Description de la structure interne et des portes dans le modèle ADC [Markopoulos 97].

La Figure 44 caractérise le modèle ADC. Seule la structure de l'interacteur est prise en compte.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phisme		Mécanism es d'adaptation		Contrôle d'exécutio	
Approches	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Resrouces	Propriétés	Préfabriqué	Génééré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
ADC																

Figure 44 : Caractérisation du modèle ADC.

La section suivante traite des travaux à dominante génétique.

### 3. Prédominance génétique

Conformément à la grille d'analyse, nous examinons ici les travaux relatifs à l'adaptation des interacteurs dans leur prise en charge de la détection du changement de contexte d'usage, du calcul de la réaction et de sa mise en oeuvre. Aucune approche, à notre connaissance, se limitant à la mise en oeuvre de la réaction, nous structurons la section en deux volets selon que l'approche traite de la détection du changement de contexte d'usage ou du calcul et de la mise en oeuvre de la réaction.

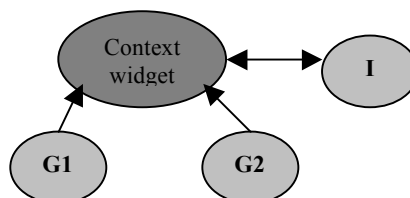
#### 3.1 Détection du changement de contexte d'usage

La Context toolkit [Dey 00] a pour objectif de faciliter le développement d'applications sensibles au contexte. Elle fournit des Context widgets dont le rôle est de mettre à disposition des applications des informations contextuelles.

Les context widgets s'appuient sur des générateurs. Les générateurs sont des composants logiciels qui encapsulent un ou plusieurs capteurs physiques. Ces capteurs assurent la perception, dans le contexte d'usage, d'informations « brutes » relatives au contexte. Les context widgets collectent ces informations et élaborent, grâce à un interpréteur, une information synthétique. Cette information est alors disponible aux applications. De façon plus précise, les context widgets permettent :

- D'exploiter des capteurs et d'abstraire une information synthétique, via des interpréteurs, à partir de données brutes ;
- D'accéder à des données contextuelles à travers le réseau et de partager ces données via une infrastructure distribuée ;
- De stocker des données contextuelles.

La Figure 45 montre l'exemple d'un context widget auquel sont associés trois composants : deux générateurs et un interpréteur. Dans cet exemple, le context widget reçoit les données contextuelles de deux générateurs G1 et G2, et sollicite l'interpréteur I pour l'élaboration d'une information synthétique. Si, par exemple, G1 et G2 fournissent respectivement le nom et le prénom d'un individu, l'interpréteur I pourra enrichir ces informations de données relatives à cet individu (sa date de naissance, son mode de vie, etc.).



*Figure 45 : Exemple de Context widget fournissant une information contextuelle calculée par l'interpréteur I en fonction des données brutes capturées par les générateurs G1 et G2. Les flèches représentent les flux de données [Dey 99].*



La Figure 46 caractérise les Context widgets sur la grille d'analyse. Cette caractérisation montre que le seul objectif des Context widgets est la détection du contexte.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq e		Valeur			Polymor phis me		Mécanis mes d'adaptation		Contrôle d'exécutio	
<b>Approches</b>	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
Context widget																

Figure 46 : Caractérisation des context widgets.

La section suivante traite des deux autres aspects : le calcul et la mise en œuvre de réaction.

### 3.2 Calcul et mise en œuvre de réaction

L'analyse de l'existant montre que les travaux portant sur le calcul et la mise en œuvre de la réaction s'appuient sur un polymorphisme des interacteurs. Dans WAHID [Jabarin 03], par exemple, les interacteurs sont des widgets graphiques classiques, dotés de plusieurs présentations et d'une capacité d'adaptation par changement de forme, dans le respect d'un certain style d'interaction. Par exemple, le menu de la Figure 47 peut passer de la forme menu déroulant (Figure 47a) à un pie menu (Figure 47b). La forme a changé, mais le style *menu* est conservé. On voit ici l'intérêt de rapprocher ces travaux de la multimodalité qui définit la notion de style d'interaction [Frohlich 91].

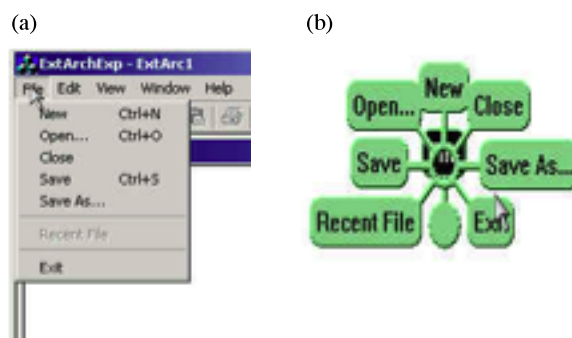


Figure 47 : Deux formes pour le même style menu [Jabarin 03].

WAHID a l'originalité de couvrir non seulement la plasticité de nouvelles applications mais aussi celle de systèmes existants. Il définit, pour ce faire, deux

architectures logicielles (dites interne et externe) permettant respectivement la construction d'IHM plastiques et la plastification de systèmes interactifs existants par l'interception et le traitement des événements d'entrée.

La Figure 48 caractérise WAHID sur la grille d'analyse. WAHID est un modèle d'interacteurs pour la plasticité. Les interacteurs WAHID s'adaptent par polymorphisme préfabriqué. Les mécanismes d'adaptation sont internes à l'interacteur.

Approches	Autodescription										Potentiel d'adaptation					
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phis me		Mécanismes d'adaptation		Contrôle d'exécutio	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Polymorphisme	Généité	Détection de la cxtx	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
WAHID												I	I	I	?	

Figure 48 : Caractérisation de WAHID [Jabarin 03].

Les metawidgets [Blattner 96] sont des widgets multimodaux. Ils s'appuient sur un répertoire de présentations ainsi que des méthodes et mécanismes de sélection pour s'afficher à l'utilisateur d'une façon optimale, c'est-à-dire limitant sa charge cognitive et respectant ses préférences. Chaque présentation exploite une ou plusieurs modalités sensorielles (visuelle et/ou auditive par exemple). Des heuristiques sont définies pour calculer la charge cognitive induite par chaque modalité ainsi que son impact sur la charge cognitive globale de l'utilisateur. La sélection d'une présentation tient compte de cette charge cognitive, des préférences utilisateur, ainsi que de facteurs propres au système.

D'un point de vue de l'architecture logicielle, les metawidgets s'appuient sur un noyau logiciel composé de deux outils : le gestionnaire de ressources et le gestionnaire de présentations (Figure 37). Ces gestionnaires sont respectivement en charge du calcul de la charge cognitive induite par chaque application et de la sélection de la présentation la plus appropriée pour un metawidget donné. Il est intéressant de noter que les présentations ne sont pas dédiées à un metawidget : elles sont partageables requérant, en conséquence, des mécanismes de consultation des présentations disponibles. Le choix de la présentation se fait à l'exécution et peut être révisé dès lors que les conditions changent.

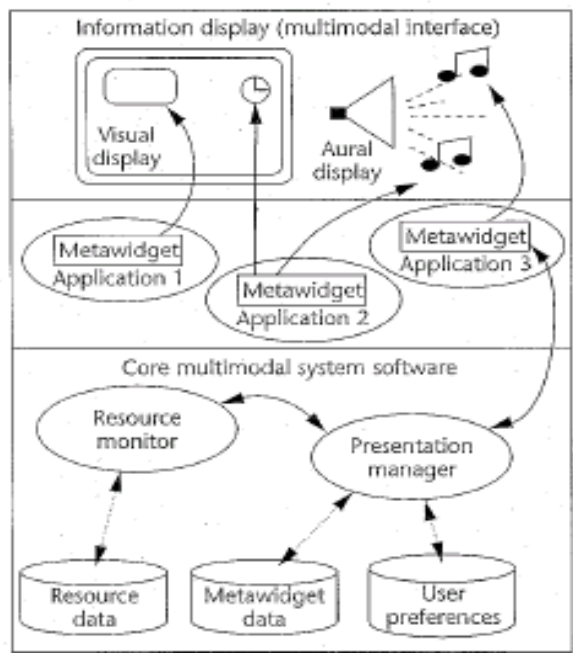


Figure 49 : Architecture logicielle pour les Metawidgets [Blattner 96].

La Figure 50 caractérise les metawidgets sur la grille d'analyse. Les metawidgets sont des interacteurs conçus pour l'adaptation. L'adaptation est assurée via le polymorphisme préfabriqué. Les mécanismes d'adaptation sont externes à l'interacteurs.

Figure 50 : Caractérisation des metawidgets [Blattner 96].

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phis me		Mécanismes d'adaptation		Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Polymorphisme	Généricité	Détection de la ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
Approches																
Metawidget										x		E	E	E	?	?

Le modèle des ressources-sensitives multimodal widgets proposé par Crease [Crease 01] définit un widget comme étant un composant qui encapsule un concept, un comportement interactif et un modèle de l'information présentée à l'utilisateur. Le widget n'est donc pas une entité totalement abstraite. Par exemple, un bouton peut être considéré comme un «sélecteur» qui encapsule le concept de «procédure», la

procédure étant activable par clic souris, mais la présentation du widget en tant qu'espace rectangulaire n'est pas spécifiée.

Le widget ne se limite pas aux interfaces graphiques. Il peut être multimodal, la modalité faisant ici référence au sens humain. Le widget est sensible aux ressources de présentation. Le terme de "ressource de présentation" dénote les moyens nécessaires à la présentation du widget. La surface d'affichage en est un exemple. Une ressource peut être disponible ou non ; convenable ou non dans un contexte d'usage donné. Par exemple, l'usage du son dans un environnement bruyant n'est pas convenable. Les caractéristiques de l'environnement sont considérées comme des contraintes sur l'usage d'une ressource de présentation. Le widget est doté d'une capacité d'adaptation.

Crease propose une architecture du widget et de la boîte à outils (Figure 51).

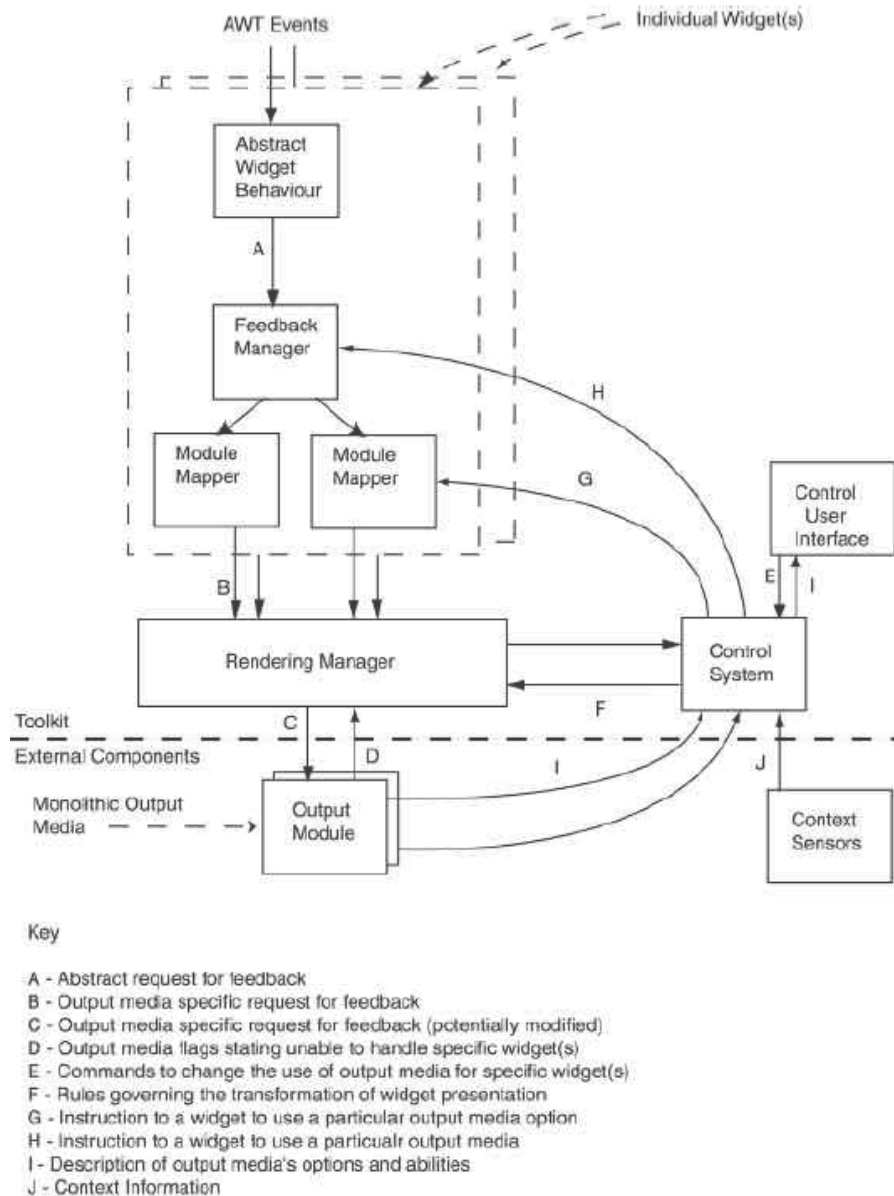


Figure 51 : Architecture de la multimodal toolkit [Crease 01].

Comme le montre la Figure 51, l'architecture du widget est définie par les composants suivants :

- l'*abstract widget behaviour* définit le comportement du widget. Il reçoit les événements en entrée et les traduit en requêtes de présentation. Lorsque l'état courant du widget change, le abstract widget behaviour sollicite un changement de présentation. La demande est traduite en termes de présentation concrète par le composant en charge des sorties : l'output module. L'abstract widget behaviour a une connaissance de l'état courant du widget mais ignore sa présentation. Les output modules ne font pas partie du widget : ils

relèvent de la boîte à outils. Un output module peut fournir une présentation dans une ou plusieurs modalités. Par exemple, un output module peut assurer un rendu graphique (une seule modalité) ; un autre un rendu audio (une seule modalité) ; mais on peut imaginer des output modules combinant plusieurs modalités (graphique et son, par exemple). Enfin, un output module peut être capable de fournir un rendu à une variété de widgets ou à un seul widget ;

- le *feedback manager* reçoit les requêtes de présentation en provenance du abstract widget behaviour. Il gère les modules mappers ;
- le *module mapper* est en interface avec les composants de la boîte à outils. Chaque widget dispose d'autant de modules mappers qu'il sollicite d'output modules.

La Figure 52 caractérise les ressources-sensitive multimodal widget sur la grille d'analyse. La préoccupation principale du modèle est l'adaptation. L'adaptation est basée sur le polymorphisme préfabriqué. Le modèle prend en compte la fonction d'un interacteur. La structure d'un interacteur est décrite par les composants abstract widget behaviour, feedback manager et module mapper et les relations entre ces composants. Les ressources-sensitive multimodal widgets sont des widgets polymorphes capables de s'adapter au changement du contexte. La capture du contexte est externe au widget. Elle est assurée par la boîte à outils. Le widget et la boîte à outil se partagent la mission de calcul de la réaction. Celle-ci est mise en œuvre par le widget.

	Fonctionnelle		Ontologique			Génétiq ue		Valeur		Polymor phisme		Mécanism es d'adaptation			Contrôle d'exécutio n	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Génére	Détection de la ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
<b>Approches</b>																
Ressources-sensitive multimodal widget	x			x						x		E	EI	I	?	?

Figure 52 : Caractérisation des ressources sensitive multimodal widgets [Crease 01].

La boîte à outil FRUIT (Flexible and Rearrangeable User Interface Toolkit) [Kawai 96] a été proposée pour surmonter les situations "de crise" pour lesquelles, par exemple, l'interface visuelle devient insuffisante ou la plate-forme dispose de fonctionnalités limitées. Typiquement, les malvoyants pourront souhaiter qu'une interface visuelle se présente sous une autre modalité. Un utilisateur peut aussi souhaiter migrer son application vers une plate-forme de moindre qualité requérant, en conséquence, une transformation de l'IHM.

FRUIT s'appuie sur une séparation classique des préoccupations : il distingue l'aspect fonctionnel du widget de sa présentation. L'aspect fonctionnel donne lieu aux Widgets Abstraits. Il identifie trois types de widgets abstraits :

- Les Widgets Abstraits de Base (AW) représentent les objets d'interaction basiques tels que les boutons ou les champs texte qui sont directement en interaction avec l'utilisateur ;
- Les Widgets Abstraits Conteneurs représentent des objets qui contiennent des objets d'interaction et en gèrent la géométrie ;
- Les Widgets Abstraits Composés représentent un groupe de widgets abstraits qui se comporte d'une façon spécifique.

En Widgets Abstraits de Base, FRUIT recense les éléments suivants :

- Commande : en graphique, cette classe peut être représentée par un bouton à condition qu'il y ait un dispositif de pointage ;
- Sélection : de même, les listes et boutons radio sont ici classiques ;
- Estimateur : il prend la forme d'échelle, typiquement le potentiomètre en graphique. En l'absence d'un dispositif de pointage, les touches directionnelles du clavier (les flèches) peuvent être utilisées. La valeur peut être aussi directement saisie au clavier ;
- Texte : le champ texte est classique en graphique, mais dès lors que les utilisateurs aveugles sont considérés, un canal auditif doit être prévu.

D'un point de vue de l'architecture logicielle, FRUIT isole l'interaction shell de l'application. Si l'utilisateur désire changer de modalité, il peut tuer l'interaction shell, en invoquer un autre et le connecter à l'application. Un session manager s'exécute sur chaque hôte. Il a pour rôle de gérer les applications FRUIT.

La Figure 53 caractérise FRUIT sur la grille d'analyse. La grille montre que FRUIT est un modèle des interacteurs pour l'adaptation. Les interacteurs FRUIT sont polymorphes à la conception. Aucune indication sur l'acteur de la détection du contexte d'usage. Le calcul et la mise en oeuvre de la réaction sont externes à l'interacteur.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phisme	Mécanismes d'adaptation		Contrôle d'exécutio		
Approches	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Coûts	Propriétés	Préfabriqué	Génére	Détection delta ctxt	Calcul de réaction	Mise en oeuvre	Gestion d'état	Gestion d'exécution
FRUIT	X									X		?	E	E	?	?

Figure 53 : Caractérisation de FRUIT [Kawai 96].

UBI (UBiquitous Interactor) [Nylander 04] cible les IHM multi-plates-formes. UBI est centré sur la notion de service. Il offre la possibilité de développer une version

unique du service, indépendante de la plate-forme et de créer des interfaces utilisateurs spécifiques pour ce service. Pour ce faire, UBI utilise les *actes d'interaction* afin de décrire l'interaction entre l'utilisateur et le service tout en étant indépendant de la plate-forme. L'interaction est exprimée en terme d'actes d'interaction. Les actes d'interaction sont des unités abstraites d'interaction entre l'utilisateur et le service. Ils couvrent à la fois les interactions en entrée et en sortie. Les actes d'interaction ne contiennent aucune information quant à la modalité et la présentation. Typiquement, "sélectionner un élément dans un ensemble", "afficher" ou "modifier une information" sont des actes d'interaction. Par contre, presser un bouton ou prononcer une commande n'en sont pas du fait de leur dépendance à une modalité. Une liste d'actes d'interaction est fournie. Elle s'appuie sur une analyse des applications et services existants. Elle identifie les actes suivants : Input, output, select, modify, create, destroy, start et stop. Les actes Input et output sont définis d'un point de vue système. Select agit sur un ensemble d'alternatives prédéfinies. Les actes create, destroy et modify représentent le cycle de vie des données d'un service particulier tandis que start et stop relèvent de la session d'interaction.

Les IHM sont générées sur la base des actes d'interaction (Figure 54). Les services sont décrits en termes d'actes d'interaction. Des interpréteurs ou générateurs d'IHM (IU) produisent l'IHM à partir de ces descriptions abstraites, pour une plate-forme donnée

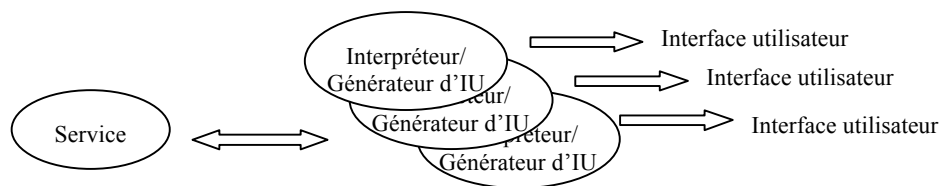


Figure 54 : Processus de production d'IHM dans UBI.

En pratique, UBI s'appuie sur trois éléments clé :

- Un langage de spécification de l'interaction (ISL pour Interaction Specification Language) pour la description en actes d'interaction ;
- Des Customisation forms pour soutenir le calcul de la présentation. Elles spécifient comment chaque service peut être présenté sur une plate-forme donnée ;
- Des Interaction engines (Figure 54) qui interprètent les actes d'interaction et génèrent leur présentation pour la plate-forme ciblée.

La Figure 55 caractérise UBI sur la grille d'analyse.



Approches	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq ue		Valeur			Polymor phis me		Mécanismes d'adaptation		Contrôle d'exécutio	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Coûts	Propriétés	Préfabriqué	Généré	Détection delta cxt	Calcul de réaction	Mise en oeuvre	Gestion d'état	Gestion d'exécution
UBI	X										X		E		X	X

Figure 55 : Caractérisation d'UBI [Nylander 04].

XWeb [Olsen 00] traite des sites Web multi-plate-formes. Il permet la génération de sites Web sous différents formats grâce à la notion de XView. Une XView est une description XML visant à :

- Spécifier les données rendues observables et/ou manipulables à l'utilisateur ;
- Etablir le lien (mapping) entre ces données et leur incarnation ;
- Fournir les ressources nécessaires à leur mise en œuvre.

Au cœur de la spécification des XViews, se trouve la notion d'interacteur. Un interacteur spécifie les types de valeurs qu'une donnée peut prendre sans fixer pour autant la technique d'interaction utilisée pour son exposition (affichage en graphique) et sa manipulation. Dans la Figure 56 par exemple, les valeurs « on » et « off » pour un jour particulier sont modélisées comme des 0 et 1. Ainsi, toute technique d'interaction qui permet le choix entre ces deux valeurs peut être utilisée. Ces techniques peuvent être des boutons radio, des menus, voire des textes à prononcer. Olsen distingue deux types d'interacteurs : les atomiques et les agrégats. Les atomiques implémentés sont les suivants : énumération, nombre, heure, date et texte. Les agrégats sont les groupes, les listes et les liens.

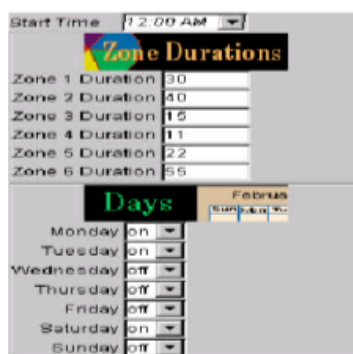


Figure 56 : Interacteur représentant un système automatique de contrôle des arroseurs à domicile.

L'adaptation à la plate-forme s'appuie sur des implémentations clientes. Le client, en fonction de ses capacités, détermine la présentation optimale de chaque interacteur.

La Figure 57 caractérise XWeb sur la grille d'analyse.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq e		Valeur			Polymor phisme		Mécanismes d'adaptation		Contrôle d'exécutio	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection de la cxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
<b>Approches</b>																
XWeb	x										x					

Figure 57 : Caractérisation de Xweb [Olsen 00].

XForms [W3C 06] est la nouvelle génération de formulaires HTML. Son principe est la séparation des préoccupations entre les données et leur présentation. Il distingue :

- Le modèle XForms pour la spécification abstraite du formulaire ;
- L'interface utilisateur XForms pour la spécification de la présentation en entrée et en sortie. XForms fournit ici une ensemble standard d'éléments visuels : les contrôles.

Cette séparation permet de concevoir les données sans se préoccuper de leur présentation. En pratique, XForms distingue trois niveaux :

- La présentation en termes de boutons, champs texte, etc. ;
- La logique pour maintenir les relations entre les champs du formulaire ;
- Les données pour maintenir l'état courant de l'application.

Avec XForms, il est ainsi possible de définir des formulaires sans présager de la plate-forme cible. Un même formulaire peut s'afficher sur PDA, téléphone portable, etc., garantissant, en conséquence, une cohérence de l'application à coûts de développement réduits. C'est la plate-forme cible qui choisira la présentation la plus appropriée compte tenu de ses ressources.

La Figure 58 caractérise les XForms sur la grille d'analyse.

	Autodescription								Potentiel d'adaptation							
	Fonctionnelle		Ontologique			Génétiq e		Valeur			Polymor phisme		Mécanismes d'adaptation		Contrôle d'exécutio	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Coûts	Propriétés	Préfabriqué	Généré	Détection de la cxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
<b>Approches</b>																
XForms			x							x			E	E	x	

Figure 58 : Caractérisation des XForms [W3C 06].

## 4. Synthèse

En synthèse, nous retenons de cette revue qu'aucune approche ne couvre le profil minimal requis (Figure 59).

Approches	Autodescription									Potentiel d'adaptation						
	Fonctionnelle		Ontologique			Génétiq ue		Valeur		Polymor phisme		Mécanismes d'adaptation			Contrôle d'exécutio	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Abstraction	Présentation	Détection delta ctxt	Calcul de réaction	Mise en oeuvre	Gestion d'état	Gestion d'exécution
Profil minimal	x	x	x	x	x	x	x	x	x							
Profil idéal	x	x	x	x	x	x	x	x	x	x	x	IE	IE	IE	IE	IE
ArtStudio	x	~						~								
M. Florins	x	x						~	~							
York																
Pisa																
ADC																
Context widget																
WAHID												I	I	I	?	
Metawidget										x		E	E	E	?	?
M. Crease	x			x						x		E	IE	I	?	?
FRUIT	x									x		?	E	E	?	?
UBI	x										x		E		x	x
XWeb	x									x						
XForms			x							x			E	E	x	

Figure 59 : Synthèse de la revue critique de l'état de l'art.

La principale faiblesse tient à l'absence de description de la valeur. Si les ressources sont une préoccupation fréquente dans les travaux traitant de l'adaptation ([Blattner 96] [Crease 01] [Kawai 96] [Nylander 04] [Olsen 00] [W3C 06]), les propriétés ne sont pas explicitement traitées. Seule, à notre connaissance, M. Florins intègre l'utilisabilité dans sa description d'un interacteur [Florins 06]. Chez [Blattner 96], l'adaptation est certes dirigée par la motivation de réduire la charge cognitive globale de l'utilisateur, mais l'interacteur n'exprime pas intrinsèquement la charge qu'il représente.

En filigrane, nous retrouvons partout la même préoccupation de séparation des préoccupations entre fonction et présentation. Nous retenons de [Blattner 96] la possibilité de capitaliser des présentations hors interacteur. Cette vision "open" est intéressante pour :

- Tirer profit de présentations nouvelles, imaginées et réalisées post-crédation de l'interacteur ;
- Ouvrir les interacteurs à des plates-formes d'interaction nouvelles, non prévues lors de l'élaboration de la boîte à outils.

En matière de fonctions, il n'existe pas de classification consensuelle. Les propositions sont nombreuses sans qu'un effort ne soit vraiment porté à leur comparaison. D'un point de vue des présentations, la majorité des travaux ciblent historiquement le graphique. Quelques approches s'ouvrent depuis peu sur d'autres modalités ([Blattner 96] [Crease 01] [Kawai 96] [Nylander 04]). Il serait, en conséquence, pertinent d'examiner les travaux en multimodalité. Enfin, nous remarquons que fonctions et présentations sont systématiquement métier. Les capacités d'adaptation de l'interacteur ne sont pas spécifiées et aucune présentation ne leur est affectée.

Le chapitre suivant propose une ouverture des recherches en IHM et hors IHM pour combler les manques de l'état de l'art en matière d'interacteurs. L'accent sera mis sur la valeur et ce que l'on peut gagner des autres disciplines en la matière.



# CHAPITRE III

## APPORTS PLURIDISCIPLINAIRES

---

La revue de l'état de l'art a révélé comme principale lacune des travaux existants la description de la valeur d'un interacteur. Les efforts ont jusqu'ici été majoritairement portés sur la description fonctionnelle d'un interacteur ainsi que son polymorphisme en termes de présentation, sans que les ressources qu'il requiert et les propriétés qu'il garantit n'aient été formulées.

Ce chapitre se propose d'élargir l'horizon en IHM et hors IHM pour recueillir un ensemble d'acquis pluridisciplinaires qui seraient ici pertinents.

Dans une première section, nous présentons, à visée structurante, un cadre de référence révisé pour la plasticité des IHM. Ce cadre de référence intègre explicitement ressources et propriétés dans la méthode de (rétro-)conception des IHM. Nous examinons ensuite différentes disciplines pour décrire respectivement ressources (section 2) et propriétés (section 3). Nous terminons par une synthèse (section 4).

---

### 1. Cadre de référence révisé pour la Plasticité des IHM

Depuis toujours en IHM, on écrit des scénarii d'usage [Rosson 02]. Leur analyse aide à identifier les contextes d'usage (<Utilisateur, Plate-forme, Environnement>) et à comprendre le domaine en termes de concepts et de tâches utilisateur. La conception de l'IHM prend alors place, se structurant autour des notions d'IHM abstraite, concrète et finale telles qu'esquissées dans l'architecture générale des Model-Based Interface Design Environment (MB-IDE) [Szekely 96] :

- Les tâches utilisateur et concepts du domaine décrivent l'interaction homme-machine selon une perspective domaine, sans préjuger, si possible, d'une quelconque représentation. Des outils tels que CTT [Paterno 97] pour les tâches et les diagrammes de classe UML pour les concepts sont ici classiques ;
- L'interface abstraite structure l'IHM en espaces de travail aussi dits espaces de dialogue ou unités de présentation. Un espace de travail est un « lieu d'activité virtuel offrant les éléments nécessaires à la réalisation d'une ou plusieurs tâches » [Normand 92] ;

- L'interface concrète fait le choix des interacteurs pour :
  - Les espaces de travail alors réifiés en fenêtres ou canevas dans le cas d'IHM graphiques ;
  - Les enchaînements entre espaces alors incarnés en séparateurs (espace ou trait) ou objets de navigation (boutons, liens hypertexte) ;
  - Les tâches élémentaires et concepts du domaine matérialisés en boutons radio, cases à cocher, images, etc. ;
- L'interface finale fait le choix d'un langage de programmation et environnement d'exécution.

Ces niveaux d'abstraction sont généraux. Aussi, ont-ils été repris en plasticité pour raisonner sur l'ingénierie et rétro-ingénierie d'IHM plastiques. Le projet européen CAMELEON en fait sa colonne vertébrale qu'il enrichit de préoccupations propres à la plasticité, à savoir [Calvary 03] :

- L'identification des contextes d'usage et changements de contexte d'usage que le système interactif a pour ambition de couvrir ;
- L'adaptation à appliquer lors de ces changements ;
- L'infrastructure d'exécution pour la mise en œuvre de l'adaptation.

De façon plus précise (Figure 60) :

- Le contexte d'usage s'exprime en termes d'utilisateur, de plate-forme et d'environnement. Chaque contexte peut justifier une conception/rétro-conception spécifiques. Sur la figure 60, deux fils d'ingénierie sont prévus pour couvrir les deux contextes d'intérêt (modèles archétypes) ;
- L'adaptation est spécifiée en termes d'évolution et de transition. L'évolution spécifie l'objectif à atteindre (par exemple, migrer lorsque la batterie faiblit), tandis que l'évolution indique un éventuel accompagnement de l'utilisateur dans le changement. Les lignes jaunes dans les surfaces augmentées [Rekimoto 99] sont un exemple de transition (Figure 61) ;
- L'infrastructure d'exécution soutient la mise en œuvre de l'adaptation. La mise en œuvre s'appuie sur une connaissance du système interactif dans son état et son histoire (modèles dits observés) ainsi que des mécanismes pour la détection du changement de contexte d'usage, le calcul et la mise en œuvre de l'adaptation (respectivement notés S, C et E sur la figure 60 pour Sensing, Computing et Executing). Ces mécanismes peuvent être généraux, assurés par l'infrastructure d'exécution et/ou spécifiques au système interactif, alors embarqués dans les interfaces finales (Figure 60).

Dans cette version (Figure 60), le cadre de référence s'inscrit fortement dans une approche à base de modèles :

- Les modèles sont conformes à des méta-modèles (à l'époque, appelés modèles ontologiques) ;
- La conception/rétro-conception d'IHM se fait par transformations cumulatives de modèles : en particulier, les modèles amont sont

conservés lors de réifications. L'interface concrète (D10), par exemple, connaît tous les modèles ayant présidé à sa production : sur l'exemple Config 2 de la Figure 60, il s'agit des modèles concepts et tâches (D8), interface abstraite (D9), plate-forme (D4) et environnement (D5). Les références à la plate-forme et l'environnement expriment ici des dépendances vis-à-vis du contexte d'usage ;

- Les modèles existent à l'exécution, donnant lieu à des instanciations  $R_i$  (pour Runtime) et non plus seulement  $D_i$  (pour Design time).

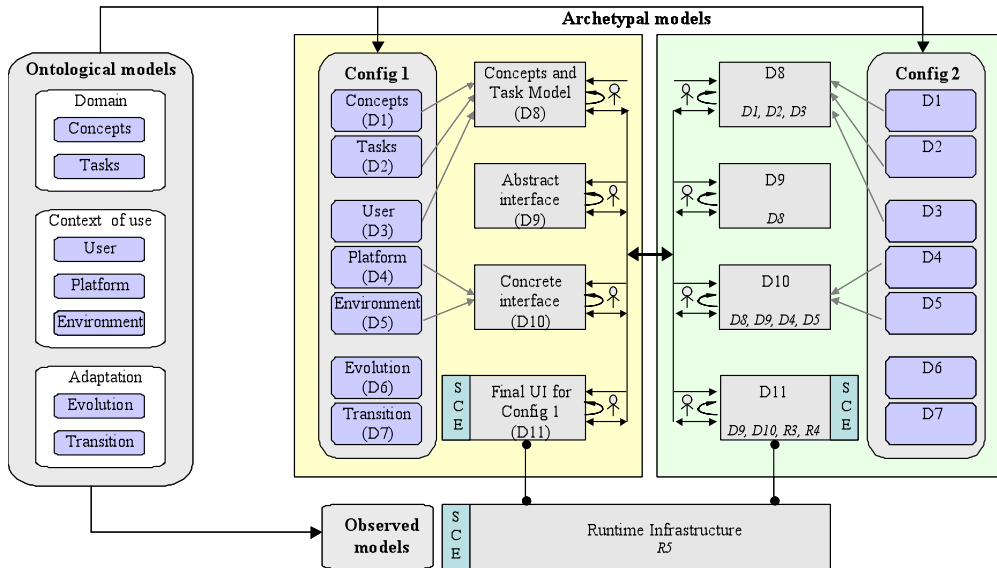


Figure 60 : Cadre de référence pour la plasticité [Calvary 03].

Dans les surfaces augmentées [Rekimoto 99] (Figure 61), une ligne jaune fournit un retour d'information à l'utilisateur quant au déplacement d'entités numériques (ici un document) dans le monde physique (une table). C'est un exemple d'IHM de transition.



Figure 61 : Surfaces augmentées [Rekimoto 99].



Rappelons que la *valeur* est définie comme "une grandeur qui croît lorsque la satisfaction du besoin de l'utilisateur augmente ou que l'ensemble des dépenses diminue [Lachnitt 80]. La notion de valeur apparaît dès lors qu'un sujet, pour fixer son choix entre plusieurs produits, établit une relation subjective entre une impulsion (un besoin) et des freins (des dépenses)" [Chevallier 89].

Si on en revient à la notion de *Valeur*, on se rend compte que le cadre de référence ne couvre que partiellement et implicitement cette notion :

- Les ressources apparaissent en filigrane par les dépendances des modèles vis-à-vis du contexte d'usage, mais l'analyse reste gros grain ;
- Les propriétés qui motivent les transformations de modèles ne sont pas explicites.

Nous proposons une version révisée du cadre de référence qui intègre explicitement les propriétés par le biais de critères qualité et de métriques associées (Figure 62). Dans ce cadre révisé :

- Le contexte d'usage fournit des entités numériques, physiques et cognitives qui pourront jouer le rôle de ressources d'interaction ;
- La qualité fournit un ensemble de critères et métriques qui joueront le rôle de référentiel pour exprimer les propriétés garanties par le système interactif.

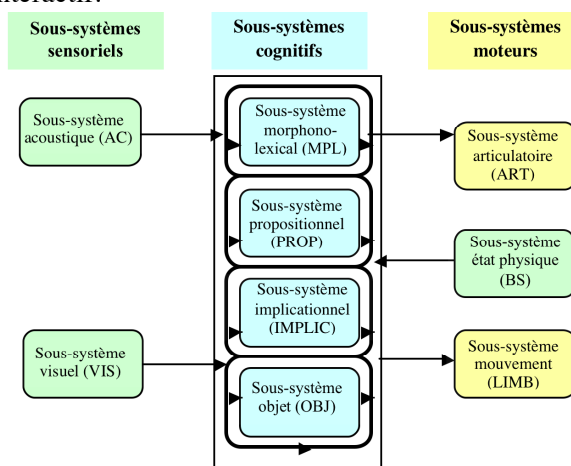


Figure 62 : Cadre de référence révisé pour la plasticité. Adapté de [Calvary 05].

Les sections suivantes examinent différentes disciplines pour affiner l'expression et la mesure des ressources et propriétés. Nous terminons par une synthèse.

—

## 2. Ressources

Nous proposons, en premier lieu, un espace de classification des ressources. Cet espace a pour vocation de servir de trame à la revue des acquis.

### 2.1 Espace de classification

Nous distinguons les travaux généraux, applicables à tout métier (fonction ou service), des travaux spécifiques traitant de l'adaptation. Pour chacun d'eux, nous examinons les acquis pour le recensement et la mesure des ressources requises/consommées. Conformément au cadre de référence révisé (Figure 62), nous distinguons les ressources selon qu'elles relèvent de l'utilisateur, de la plate-forme et de l'environnement (Figure 63). Elles ont toutes une empreinte physique, à compléter pour :

- L'utilisateur : de requis cognitifs ;
- La plate-forme : de requis numériques ;
- L'environnement : de conditions sociales.

La Figure 63 présente notre espace de classification. Nous distinguons successivement :

- Ressources utilisateur : comprennent des ressources physiques et cognitives. Nous désignons par ressources physiques les capteurs et effecteurs humains tels que les yeux qui captent l'information ou le doigt qui agit sur un objet physique. Les ressources cognitives désignent les ressources mentales nécessaires pour le traitement de l'information.
- Ressources de la plate-forme : les ressources physiques de la plate-formes désignent les dispositifs d'entrée et de sortie. Les ressources numériques représentent les composants de traitement de l'information de l'ordinateur tels que la mémoire ou le processeur.
- Ressources de l'environnement : dans leurs aspect physique, les ressources de l'environnement représentent les caractéristiques physiques telles que la pression atmosphérique, température...etc. Nous identifions aussi les ressources sociales de l'environnement.

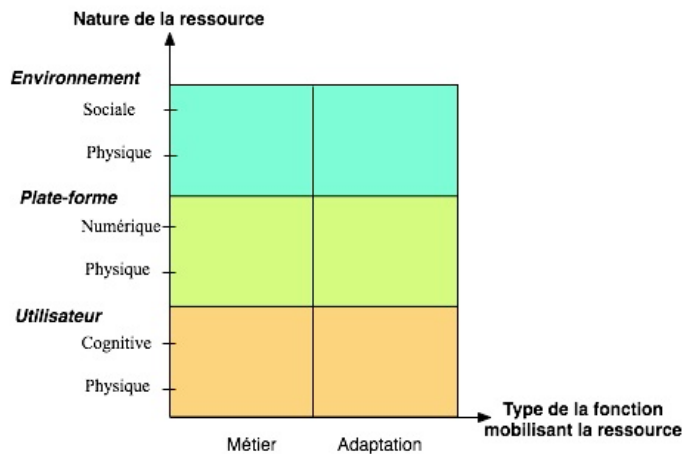


Figure 63 : Espace de classification des ressources.

La section suivante recense les acquis applicables pour la qualification et quantification des ressources mobilisées en interaction. Le recensement sera fait selon les deux types de fonction : métier et d'adaptation.

## 2.2 Apports pluridisciplinaires

Cette section examine, en premier lieu, les acquis généraux puis traite spécifiquement de l'adaptation.

### 2.2.1 Métier

Un même objet d'étude qu'est l'Interaction Homme-Machine peut être examiné selon une perspective utilisateur versus système, donnant lieu à des modèles et théories à dominante cognitive versus technique. Nous structurons la revue selon ces deux angles d'étude. Nous terminons par la perspective environnementale qui peut avoir une incidence sur l'utilisateur et la plate-forme.

### Perspective utilisateur

Les travaux ici recensés adoptent une perspective utilisateur dans l'analyse de l'interaction Homme-Machine. En termes du cadre de référence révisé (Figure 62), ils portent sur l'utilisateur tel que pressenti à la conception et/ou observé à l'exécution (Figure 64).

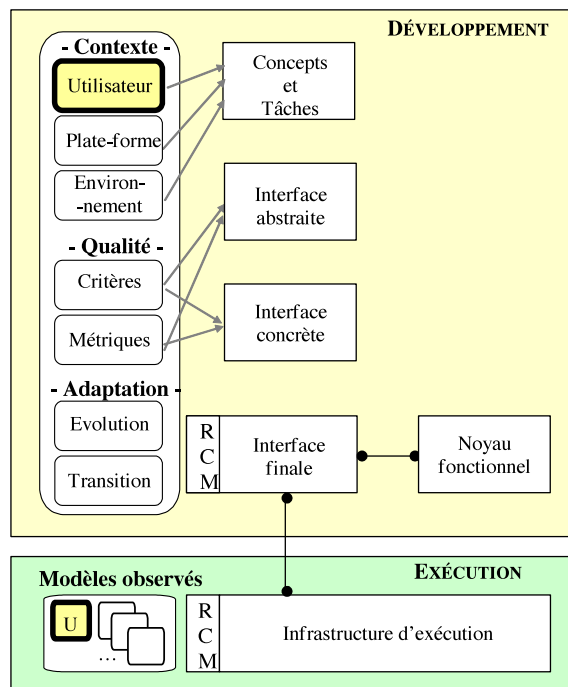


Figure 64 : Localisation des modèles de l'utilisateur dans le cadre révisé de la plasticité.

Nous recensons ci-dessous les acquis puis en tirons quelques enseignements pour l'identification et la mesure des ressources physiques et cognitives.

### ➤ **Modèle du processeur humain**

Le modèle du processeur humain [Card 83] appréhende le sujet humain dans sa globalité et adopte, dans ses descriptions, la terminologie de l'informaticien. Il assimile l'utilisateur à un système de traitement de l'information constitué de trois sous-systèmes (sensoriel, moteur et cognitif). Chaque sous-système est dimensionné en termes de mémoire (capacité, persistance, type d'information) et de processeur (temps de cycle).

Plusieurs lois y trouvent place, comme :

- La loi de Fitts qui prédit que le temps mis pour atteindre une cible est proportionnel à la distance main-cible et inversement proportionnel à la taille de la cible. La formule est  $T = a + b.log_2(D/L + 1)$  où :
- $T$  est le temps moyen mis pour effectuer le mouvement ;

- $a$  et  $b$  sont des constantes empiriques ;
- $D$  est la distance séparant le point de départ du centre de la cible ;
- $L$  est la largeur de la cible ;
- La loi de Hick-Hyman qui mesure le temps mis par un utilisateur pour prendre une décision en fonction du nombre de choix mis à sa disposition. Etant donné  $n$  choix équiprobables, le temps moyen de réaction  $T$  requis pour choisir parmi eux est approximativement :  $T = b \cdot \log(n+1)$  où  $b$  est une constante ;
- La loi de l'expérience (Power law of practice) qui mesure l'effet de fréquence sur la réalisation d'une tâche. Plus on pratique une tâche, plus le temps nécessaire à sa réalisation diminue. La loi est donnée par la formule :  $T = aP^b$  où  $T$  est le temps de réalisation de la tâche,  $P$  est la fréquence de pratique,  $a$  et  $b$  sont des constantes.

De ce modèle, nous retenons :

- La possibilité d'affinement des ressources selon la fonction de perception (sous-système sensoriel), d'action (sous-système moteur) ou de cognition qui les mobilise ;
- L'ancrage dans une logique de performance pour valider un modèle : la mesure est le temps.

## ➤ ICS

Dans la lignée du modèle du processeur humain, le modèle ICS (Interacting Cognitive Subsystems) [Barnard 85] décompose le sujet humain en sous-systèmes interconnectés (Figure 65) :

- Les sous-systèmes perceptuels acquièrent l'information en provenance du corps (état physique) et de l'environnement (visuel et acoustique). Ils transmettent ces informations aux sous-systèmes centraux ;
- Les sous-systèmes centraux (objet, morphono-lexical, propositionnel et implicationnel) réalisent des transformations de l'information acquise. Ces sous-systèmes sont au cœur des processus d'interprétation et de compréhension, ainsi que des mécanismes de raisonnement. Ils sont reliés entre eux par un réseau grâce auquel ils peuvent communiquer. Ils pilotent les sous-systèmes moteurs ;
- Les sous-systèmes moteurs (articulatoire et mouvement) contrôlent les actions physiques.

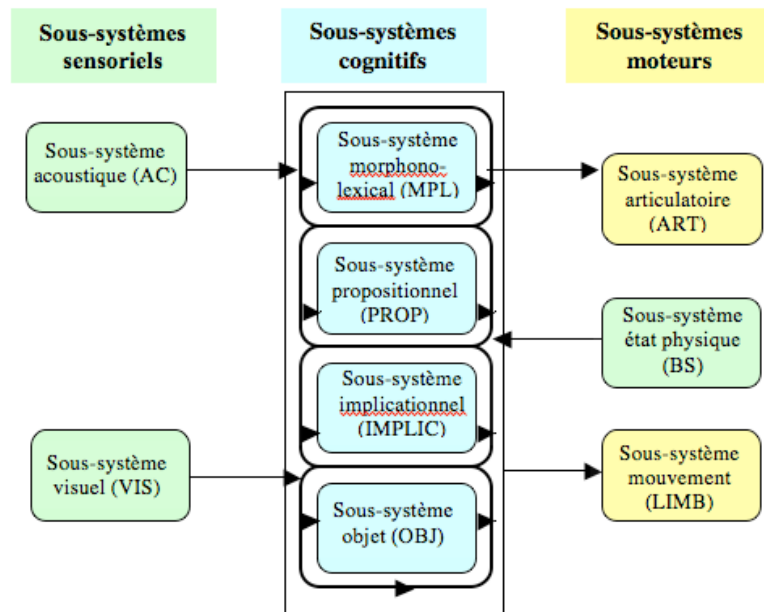


Figure 65 : Architecture générale du modèle ICS. La partie centrale constitue un réseau permettant aux sous-systèmes de communiquer.

Ces sous-systèmes fonctionnent en parallèle et sont tous modélisés selon une même structure : ils sont constitués d'entrées et de sorties, d'une capacité de traitement de l'information, et d'une mémoire locale (Figure 66). Chaque sous-système opère dans un format (code) qui lui est propre, requérant, en conséquence, des transformations avant transmission aux autres sous-systèmes.

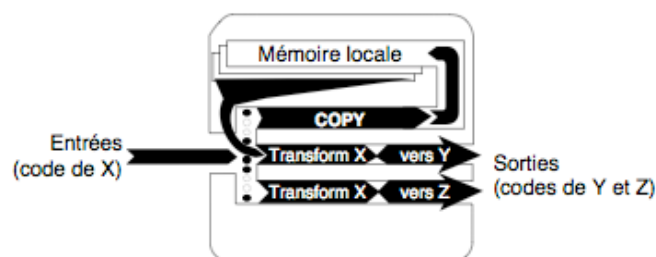


Figure 66 : Structure d'un sous-système dans ICS. Le sous-système X reçoit en entrée des informations exprimées dans son code et les transforme dans les formats attendus par les sous-systèmes Y et Z avec lesquels il communique. Extrait de [Salber 95].

On trouvera dans [Salber 95] une description détaillée d'ICS. En voici un extrait : « En règle générale, chaque sous-système perceptif a un partenaire privilégié parmi les sous-systèmes centraux. Par exemple, OBJ pour VIS et MPL pour AC. OBJ et MPL abstraient l'information du niveau perceptif vers un niveau intermédiaire qui permettra ensuite l'interprétation des informations perçues.

Le sous-système objet (OBJ) peut être vu comme ayant la charge de “l’imagerie mentale”. C’est là par exemple que se forment les images mentales qui ne sont pas directement perçues mais qui sont générées par le système cognitif. Ce sous-système va en général, à partir des représentations fournies par VIS, élaborer des propositions, c’est-à-dire des relations sémantiques entre entités, à l’intention du sous-système PROP. Cette transformation correspond à l’identification d’un objet et à l’établissement de relations avec les objets qui l’entourent (par exemple, l’objet vu est un stylo et il est posé sur une table). Un cas particulier notable est celui de la perception du texte écrit : dans ce cas, OBJ va élaborer une représentation à l’intention du sous-système MPL qui est en charge du langage.

Le sous-système morphono-lexical (MPL) est lié principalement au langage. On peut décrire les représentations qu’il manipule comme “ce que nous entendons dans notre tête”. Ce sous-système a la connaissance des mots et des formes lexicales. Les représentations fournies à ses entrées par AC pour le langage parlé ou par OBJ (après transformation VIS  $\Rightarrow$  OBJ) pour le langage écrit sont transformées en propositions pour le sous-système PROP. Cette transformation extrait le sens des formes langagières entendues ou lues. C’est aussi MPL qui permet la génération de formes langagières. Pour cette génération, le sous-système PROP va transformer une proposition en une représentation pour MPL. Cette représentation va ensuite être transformée par MPL à destination du sous-système articulatoire qui va l’exprimer sous forme de langage parlé ou écrit.

Le sous-système propositionnel (PROP) manipule des représentations correspondant à des propositions. Ces propositions peuvent être représentées sous forme de prédicats logiques. Les entrées de PROP sont principalement alimentées par les sous-systèmes OBJ, MPL et IMPLIC. Les représentations circulant dans PROP correspondent au contenu sémantique des informations perçues par les sous-systèmes perceptifs mais aussi des propositions engendrées par IMPLIC. Le sous-système PROP a pour partenaire privilégié le sous-système IMPLIC.

IMPLIC reçoit en entrée des représentations qui proviennent principalement de PROP et génère de nouvelles représentations à l’intention de PROP. Pour simplifier et très schématiquement, on peut assimiler le couple PROP/IMPLIC à un système expert où PROP serait la base de connaissances et où IMPLIC jouerait le rôle du moteur d’inférence. Le cycle PROP  $\Rightarrow$  IMPLIC et IMPLIC  $\Rightarrow$  PROP modélise le raisonnement tel que l’entend le sens courant. Le sous-système PROP génère aussi des représentations dans le code des sous-systèmes OBJ et MPL qui vont nous permettre d’agir sur l’environnement par l’intermédiaire des sous-systèmes moteurs.

Les deux sous-systèmes moteurs contrôlent nos actions sur le monde physique. Le sous-système articulatoire (ART) est spécialisé dans l’émission du langage qu’il soit articulé, mais aussi écrit ou tapé au clavier. Le sous-système mouvement (LIMB) se charge de tous les autres mouvements du corps. Notons qu’en règle générale, le sous-système LIMB reçoit des représentations du sous-système objet, traduisant le fait que nous préparons visuellement un geste avant de l’effectuer. Toutefois dans le cas particulier des actes réflexes (par exemple, on retire sa main rapidement lorsque l’on touche un objet brûlant), le sous-système état physique communique directement avec le sous-système mouvement. C’est l’un des cas où deux sous-systèmes peuvent

communiquer sans mettre en jeu un sous-système central (un autre cas survient dans la même circonstance : on criera "Aïe!", ce qui fera aussi intervenir une communication directe BS  $\Rightarrow$  ART) » [Salber 95].

Nous retenons de ce modèle :

- La possibilité d'affiner chaque fonction (perception, action, cognition) selon les sous-systèmes correspondants ;
- Les transformations de code qui accompagnent chaque changement de sous-système ;
- Le caractère potentiellement itératif du traitement, alourdissant, en conséquence, le coût cognitif de la tâche.

### ☛ Théorie de l'Action

Contrairement aux modèles cognitifs précédents (Modèle du processeur humain et ICS), il ne s'agit pas ici de modéliser structurellement le sujet humain. La Théorie de l'Action [Norman 86] précise les étapes cognitives mises en œuvre dans la réalisation d'une tâche. En point de départ, la Théorie de l'Action suppose l'établissement d'un but pour l'utilisateur. Ce but est une représentation mentale de l'état que souhaite l'utilisateur pour le système. Il se formule en termes de variables psychologiques faisant sens pour l'utilisateur : par exemple, dans le domaine de l'édition de documents, les notions de mots ou de paragraphes. Le but peut, par exemple, consister à disposer d'un paragraphe contenant un seul mot : le terme "Illustration". La Théorie de l'Action identifie les étapes mises en œuvre entre l'établissement de ce but et l'évaluation de son atteinte par l'utilisateur. Ces étapes s'articulent en deux phases, l'exécution et l'évaluation (Figure 67), ici illustrées sur l'édition de document.

L'exécution comporte trois étapes :

- La *formation d'une intention*. L'intention vise à abolir la distance entre le but et la représentation mentale que l'utilisateur se fait de l'état courant du système. La formation de l'intention requiert une traduction entre les variables physiques du système et psychologiques du but. Supposons ici que le paragraphe contienne la phrase "Illustration de la Théorie de Norman". Etant donné le but, l'intention consiste à détruire les mots " de la Théorie de Norman" ;
- La *spécification d'un plan d'actions* pour atteindre le but. Cette spécification requiert une traduction de l'intention psychologique en variables physiques. Elle prescrit ici de sélectionner puis couper la séquence " de la Théorie de Norman" ;
- L'*exécution* du plan d'actions pour modifier l'état du système. L'exécution consiste ici, de façon effective, à sélectionner puis couper la séquence " de la Théorie de Norman". Le retour d'information donné par le système est ensuite évalué par l'utilisateur pour savoir si son but est atteint. C'est la partie évaluation.



L'évaluation se décompose symétriquement en trois étapes :

- La *perception* du retour d'information donné par le système suite à l'exécution du plan d'actions. L'utilisateur constate ici que la séquence " de la Théorie de Norman" a disparu ;
- L'*interprétation* de cette perception en variables psychologiques. Cette interprétation requiert une traduction entre les mondes physique et psychologique. Elle conclut ici à la destruction de la séquence " de la Théorie de Norman" ;
- L'*évaluation* du nouvel état du système au regard du but fixé. L'utilisateur constate ici l'obtention d'un paragraphe contenant un seul mot : "Illustration". Son but est donc atteint. Dans le cas contraire, la théorie stipulerait le déploiement d'un nouveau cycle "Exécution-Evaluation".

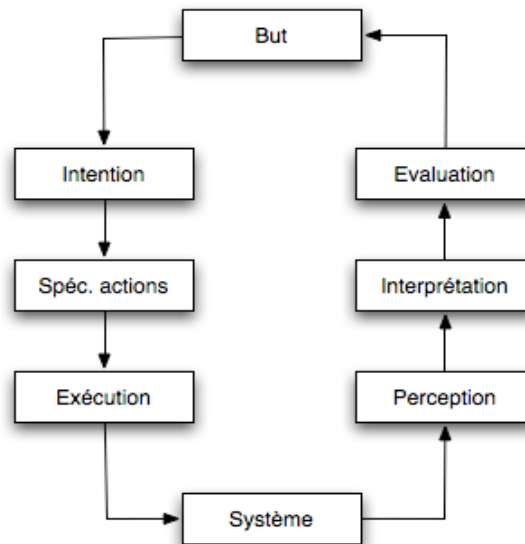


Figure 67 : Théorie de l'Action [Norman 86].

De cette théorie, nous retenons :

- Les cycles d'exécution/évaluation ainsi que leurs étapes ;
- Les notions de distances d'exécution et d'évaluation représentant les écarts entre les mondes psychologiques et physiques. Ces distances vont engendrer un effort cognitif de mise en correspondance entre la représentation mentale interne de l'utilisateur et l'image physique du système.

Ces trois modèles permettent de raisonner sur les ressources physiques et cognitives.

#### **a) Ressources physiques**

Le corps est engagé dans l'interaction. Il fournit aux sous-systèmes :

- Sensoriels : des capteurs. Par exemple, les yeux, les oreilles, le nez, la peau, etc. ;
- Moteurs : des effecteurs tels que les doigts classiquement ou encore les cordes vocales ;
- Cognitifs : des capacités de traitement : le cerveau en l'occurrence.

Les alternatives se situant aux niveaux sensoriel et moteur, ce sont ces ressources physiques qu'il sera pertinent de recenser.

L'utilisation d'une ressource engendre un effort physique. La fatigue, le rythme cardiaque, etc., en sont des indices. Si l'on s'appuie sur le modèle Keystroke [Card 83] pour les entrées, on peut supposer que l'effort intègre :

- Le couplage entre l'effecteur humain (par exemple, la main) et le capteur système (par exemple, la souris). Ceci comprend pour le couplage main-souris les efforts physiques pour, d'une part, atteindre le capteur système (la souris), d'autre part, le saisir ;
- L'exploitation du couplage pour la réalisation de la tâche. L'effort physique sera d'autant plus important que le nombre d'actions physiques sera grand ;
- Le découplage.

Il est tentant de raisonner symétriquement pour les sorties et d'intégrer les efforts physiques de couplage, exploitation et découplage.

## **b) Ressources cognitives**

[Halford 93] distingue ressources, capacités et charges cognitives :

- Les *ressources cognitives* renvoient à l'énergie mentale disponible pour un individu donné à un instant donné pour une classe particulière de traitements. Les ressources diffèrent d'un individu à l'autre et varient, en outre, dans le temps pour un individu donné ;
- La *capacité cognitive* désigne la quantité maximale de ressources qu'un individu peut mobiliser. C'est son pouvoir cognitif. Elle varie d'un individu à l'autre mais reste constante dans le temps pour un individu donné, hors mis bien entendu lors de changements physiologiques tels que la maturation, la maladie, le vieillissement, etc. ;
- La *charge cognitive* n'est pas une propriété de l'individu mais de la tâche. Elle représente le niveau d'effort mental (la quantité de ressources) nécessaire à la réalisation d'une tâche donnée.

L'effort mental [Barouillet 96] correspond au coût cognitif. Il est dépendant de trois paramètres :

- Le niveau d'expertise du sujet. Par exemple, [Olive 97] trouve que pour une tâche de rédaction, l'effort cognitif est fonction du niveau d'habileté des sujets en production écrite de texte. Les rédacteurs

moins experts utilisent plus de ressources attentionnelles que les experts ;

- Le niveau de développement ;
- La stratégie adoptée pour résoudre la tâche. Deux stratégies sont, par exemple, possibles pour calculer le produit de 37 par 98 :
  - Calculer  $37 \times 90$  puis  $37 \times 8$  puis additionner les deux résultats ;
  - Calculer  $37 \times 100$  puis  $37 \times 2$  et soustraire ce deuxième nombre au premier.

A niveaux d'expertise et de développement équivalents, la deuxième stratégie est toujours plus économique que la première.

Dans la théorie de la charge cognitive (Cognitive load theory), Sweller [Sweller 91] propose des recommandations pour réduire la charge cognitive. On retient la nécessité de :

- Réduire la redondance et les informations répétitives ;
- Utiliser les représentations visuelles et auditives lorsque ceci est nécessaire sans que ces deux modalités ne soient utilisées de façon redondante.

Il est intéressant de noter que le modèle ICS permet d'expliquer ces deux recommandations :

- Chaque information perçue mobilise des sous-systèmes cognitifs qui transforment et s'échangent l'information. En conséquence, réduire le nombre d'informations devrait effectivement réduire la charge cognitive ;
- Les informations visuelles et auditives mobilisent des sous-systèmes perceptuels spécifiques qui communiquent avec des sous-systèmes cognitifs privilégiés (OBJ pour VIS et MPL pour AC). En conséquence, réduire l'utilisation conjointe de modalités, c'est certainement réduire la charge cognitive en aval.

En synthèse, nous retiendrons qu'il n'existe pas de formule de calcul de l'effort cognitif. Plusieurs variables entrent en jeu. Il existe deux types de méthodes de mesure : les mesures subjectives et objectives. Les premières se basent sur des questionnaires. Pour l'évaluation de la charge de travail, nous citons les questionnaires NASA TLX (Task-Load-Index) [Hart 88]. Elles consistent à mettre les utilisateurs en situation d'interaction puis leur poser des questions permettant d'évaluer a posteriori l'effort cognitif engagé dans la réalisation de la tâche. Pour une mesure objective de l'effort cognitif, les psychologues passent par des variables plus sûres. Il s'agit, par exemple, de mesurer les battements du cœur de l'utilisateur lorsqu'il est en train d'utiliser le système, la fréquence du mouvement des paupières, ou encore la tension du sang. Dans les deux cas (évaluations subjectives et objectives), l'effort cognitif est toujours dépendant de facteurs tels que le niveau d'expertise de l'utilisateur avec la technologie utilisée ou la tâche à réaliser, ses préférences, son état psychique à l'instant de mesure, etc.

## Perspective système

La perspective système prend la plate-forme comme angle d'étude. La Figure 68 localise les travaux correspondants dans le cadre révisé de la plasticité.

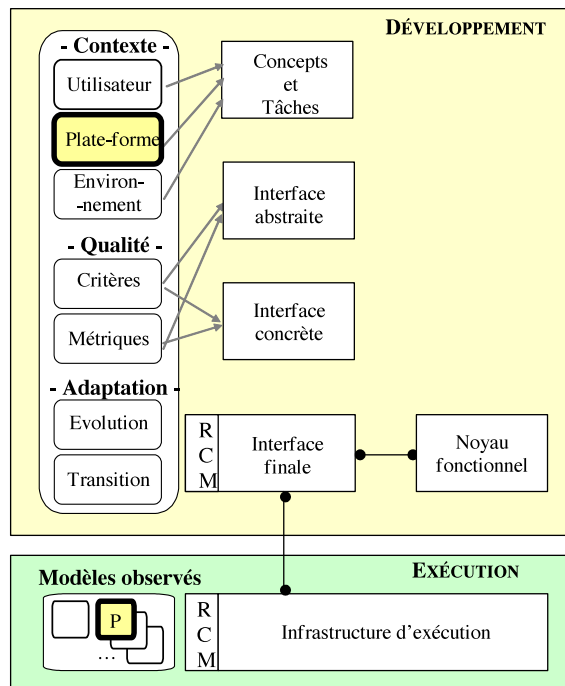


Figure 68 : Localisation des modèles de la plate-forme dans le cadre révisé de la plasticité.

CC/PP (Composite Capabilities/Preference Profiles) [W3C 06] typiquement relève de cette catégorie. La plate-forme y est décrite à quatre niveaux : a) matériel, b) logiciel, c) navigateur et d) réseau. Un exemple en est donné sur la Figure 69.

- Au niveau matériel, ce sont les caractéristiques physiques de la plate-forme qui sont spécifiées. Ces caractéristiques portent aussi bien sur les dispositifs d'entrée/sortie que les capacités de calcul de la plate-forme. Les attributs signalés sont les suivants :
  - Le nombre de bits par pixel ;
  - L'affichage couleur ou non ;
  - Le nom et le modèle du processeur ;
  - L'affichage ou non d'images ;
  - Les caractères supportés (par exemple, US-ASCII, ISO-8859 ;
  - Le type de clavier supporté (Qwerty, phonepad, etc.) ;
  - La résolution du pointage (caractère, ligne, pixel) ;
  - La taille d'écran en pixels ;
  - La sortie sonore ou non via des hauts parleurs, casque à écouteurs ou autres dispositifs de sortie sonore ;

- L'entrée de texte alphanumérique (lettres et digits) ou non (digits) ;
- L'entrée vocale ou non ;
- Au niveau logiciel, la description porte sur le système d'exploitation et les logiciels installés. On y trouve les éléments suivants :
  - Machine virtuelle java ou non, et son éventuelle version ;
  - Nom et version du système d'exploitation ;
- Le navigateur est, en particulier, décrit par son nom (par exemple, Mozilla, Internet Explorer, etc.) ;
- Au niveau réseau, on trouve des informations telles que :
  - Le serveur sur lequel la session courante est ouverte ;
  - Les serveurs supportés (par exemple, GUTS, SMS, CSD, USSD) ;
  - Le type de sécurité supporté (par exemple, PPTP).

```

[ex:LeProfil]
|
+--ccpp:component-->[ex:MaterielFinal]
|
|           |
|           +--rdf:type----> [ex:PlateformeMateriel]
|           +--ex:displayWidth--> "320"
|           +--ex:displayHeight--> "200"
|
+--ccpp:component-->[ex:LogicielFinal]
|
|           |
|           +--rdf:type----> [ex:PlateformeLogiciel]
|           +--ex:name-----> "EPOC"
|           +--ex:version--> "2.0"
|           +--ex:vendor---> "Symbian"
|
+--ccpp:component-->[ex:NavigateurFinal]
|
|           |
|           +--rdf:type----> [ex:AgentUtilNavigateur]
|           +--ex:name-----> "Mozilla"
|           +--ex:version--> "5.0"
|           +--ex:vendor---> "Symbian"
|           +--ex:htmlVersionsSupported--> [ ]
|
|           |
|           +-----+
|           |
|           +--rdf:type----> [rdf:Bag]
|           +--rdf:_1-----> "3.2"
|           +--rdf:_2-----> "4.0"

```

Figure 69 : Un exemple de profil CC/PP décrivant la plate-forme aux niveaux matériel, logiciel et navigateur Web.

CC/PP est un premier pas. Il reste certes classique dans les plates-formes décrites (clavier-écran), mais a au moins l'intérêt d'élever le drapeau de la nécessaire et, si possible normalisée, description des plates-formes. Il serait maintenant intéressant de considérer d'autres dispositifs et modalités. Par exemple, [Buxton 83] propose une

taxonomie de dispositifs basée sur leurs propriétés physiques à savoir la position, le mouvement, la pression. D'autres enrichissent la description physique d'informations relatives aux tâches pour lesquelles le dispositif serait indiqué. C'est le cas typiquement de [Foley 84] qui fait une correspondance explicite entre dispositifs et tâches. [Jacob 96] innove par de nouvelles modalités telles que les mouvements du corps, la position des pieds, de la tête, etc. via des capteurs 3-D, la voix, des dispositifs de réalité virtuelle tels que les gants par exemple.

[Hinckley 04] se démarque en considérant aussi bien les dispositifs d'entrée que de sortie. Il identifie une classe de dispositifs de pointage dans laquelle il regroupe souris, boule roulante, tablette, stylet, manette, ou encore la capture du mouvement de la tête, des pieds ou des yeux. Pour les dispositifs de sortie, il couvre les écrans, l'audio et l'haptique. En revanche, il ne fournit pas de modèle général de dispositif.

### **Perspective environnementale**

Il n'existe pas, à notre connaissance, de modèle général d'environnement. Nous considérons deux volets de l'environnement : physique et social. L'environnement physique dénote les conditions physiques telles que la luminosité, la température ambiante...etc. Dans le volet social nous regroupons les aspects culturels et sociaux.

Il est clair que ces conditions lumineuses, sonores, sociales, etc liées à l'environnement ont une incidence sur l'interaction. Typiquement, en sortie, la lisibilité (critère de Guidage/lisibilité [Bastien 93] en dépend. L'environnement aura une incidence sur les propriétés ergonomiques (lisibilité) et écologiques (respect des convenances sociales) de l'interaction.

Après ce tour d'horizon d'acquis généraux, nous recensons dans la section suivante les acquis traitant spécifiquement de l'adaptation.

### **2.2.2 Adaptation**

L'adaptation est une réaction au changement. Elle est mise en œuvre selon un processus en trois étapes consistant à (1) suspendre l'éventuelle tâche en cours ; (2) traiter le changement, puis (3) reprendre l'éventuelle tâche mise en attente. Il peut donc être pertinent d'examiner les travaux relatifs à l'interruption.

L'interruption est définie par [Oulasvirta 04] comme étant : « un événement discret pendant lequel l'attention est redirigée vers le traitement d'autres informations qui ne sont pas relatives à la tâche en cours ». Une interruption casse, en conséquence, par nature, la « continuité » du traitement de la tâche principale en forçant l'utilisateur à la suspendre momentanément. Rediriger l'attention vers une autre tâche requiert l'activation de modules relatifs à cette nouvelle tâche et l'inhibition des modules précédents. Ainsi, l'interruption provoque non seulement l'oubli (omissions) mais aussi des distorsions (intrusions) dans les représentations de la tâche principale : l'interruption n'est pas qu'un problème de temps et de frustration. Pour schématiser,

l'utilisateur va, après interruption, chercher l'information à la mauvaise place, au mauvais moment et d'une mauvaise façon, affectant en conséquence la reprise de sa tâche. Une période de récupération est nécessaire [Kahneman 73]. Chaque période de récupération exige un temps additionnel et augmente potentiellement le taux d'erreurs.

C'est ainsi que le principe des « Modalités non réservées » a émergé. L'astuce consiste à recourir à des modalités non sollicitées dans la tâche en cours. Par exemple, pour la conduite de voiture où la modalité visuelle est centrale, il sera préférable de recourir à un message audio (et non visuel) de façon à ne pas entrer en conflit avec la tâche d'orientation. Il est intéressant de noter que le modèle ICS permet d'expliquer cette propriété.

[Spence 01] chiffre le coût de changement de modalité perceptuelle. Il mesure le temps de passage d'une modalité à une autre (éventuellement d'une modalité à plusieurs autres ou de plusieurs modalités à une seule) (Figure 70). Il montre ainsi que le passage le moins coûteux est celui de la modalité auditive vers la modalité visuelle (55 ms). Le plus coûteux est celui entre le toucher et le visuel (104 ms). Ces résultats seront pertinents en plasticité pour choisir la meilleure réaction.

<b>Modalité courante</b>	<b>Modalité présentée</b>	<b>Coût en ms</b>
Auditive	Visuelle	55
Toucher	Visuelle	104
Toucher	Auditive	98
Visuelle	Auditive	68
Auditive	Toucher	67
Visuelle	Toucher	66

*Figure 70 : Coût de changement de modalité.*

[Speier 97] caractérise l'interruption par sa fréquence, sa durée, le contenu, sa complexité et son moment. Le moment de l'interruption peut être un facteur aggravant. Aussi, les instants et/ou états d'interaction pour lesquels l'interruption serait acceptable devraient-ils être spécifiés. Enfin, Schopenhauer disait que : « ce qui est important pour l'homme, ce n'est pas tant ce qui lui arrive mais ce qu'il pense qu'il lui arrive ». Aussi, le changement et le processus d'adaptation devront-ils être observables pour accompagner l'utilisateur dans le changement. L'IHM de l'adaptation sera, en conséquence, un élément clé. Les lignes ancrées de [Rekimoto 99] en sont un exemple (Figure 61). Elles sont dirigées par la propriété de continuité de l'interaction. Les propriétés font l'objet de la section suivante.

—

### 3. Propriétés

Nous proposons d'abord un espace de classification (section 3.1) puis examinons les apports potentiels permettant de raisonner sur les propriétés.

#### 3.1 Espace de classification

La littérature distingue classiquement les propriétés *fonctionnelles* et *non fonctionnelles*. Les propriétés fonctionnelles ont trait aux services rendus, c'est-à-dire, en IHM, aux tâches utilisateur et concepts du domaine manipulés. Les propriétés non fonctionnelles qualifient le service rendu en en précisant, par exemple, la performance ou la tolérance aux fautes [Chung 91] [Rosa 02]. En pratique, la frontière est mince, les propriétés non fonctionnelles se répercutant souvent par l'implémentation de fonctions assurant ces propriétés.

Qu'elles soient fonctionnelles ou non fonctionnelles, les propriétés peuvent être distinguées selon leur portée métier ou adaptation.

Notre espace de classification intègre ces deux critères discriminants (Figure 71). Sa vocation est de structurer la revue des acquis. Nous examinons ci-dessous les apports fonctionnels et non fonctionnels pour les parties métier puis adaptation.

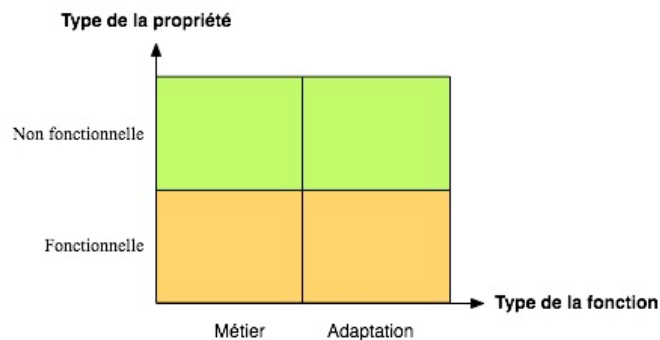


Figure 71 : Espace de classification des propriétés.

#### 3.2 Apports pluridisciplinaires

Nous examinons d'abord les acquis généraux puis ceux traitant spécifiquement de l'adaptation.

##### 3.2.1 Métier

Nous examinons ici les aspects fonctionnels puis non fonctionnels.

##### Perspective fonctionnelle

Les taxonomies de tâches sont un problème permanent dans la littérature. [Foley 84] distinguait déjà à l'époque les tâches de sélection, positionnement, etc. Des



taxonomies spécifiques à certains ‘métiers’ ont ensuite été proposées comme [Schneiderman 96] par exemple pour la visualisation des données. Il identifie sept types de données (1-dimension, 2-dimensions, 3-dimensions, multi-dimensionnel, temporel, arbre, réseau) et sept types de tâches :

- Observer la globalité de l’espace ;
- Zoomer ;
- Filtrer ;
- Obtenir des détails à la demande ;
- Relier les informations ;
- Accéder à l’historique ;
- Extraire une partie des informations.

Cette liste a ensuite été adaptée par [Pérez-Quiñones 01] pour couvrir la modalité vocale. La nouvelle liste couvre aussi les tâches de navigation sur le Web. On y trouve les tâches suivantes :

- Se situer dans l’espace d’information ;
- Naviguer ;
- Rechercher ;
- Obtenir des détails sur un élément donné.

D’autres métiers ont été traités. Par exemple, [Rudisill 88] (cité dans [Boy 89]) propose une taxonomie de tâches pour les éditeurs de dessin et de texte. Il identifie des modes (par exemple, le mode plan) pour lesquels il recense des tâches (par exemple, sélectionner, formuler et projeter pour le mode plan).

L’absence de consensus est flagrante. Aussi, l’approche par ontologies et alignement d’ontologies semble-t-elle raisonnable. Nous retenons néanmoins de [Thevenin 01] la distinction entre tâches génériques (qu’il dit d’utilité publique) et spécifiques, c’est-à-dire propres à un métier donné.

### **Perspective non fonctionnelle**

Très couramment, on utilise le terme de qualité comme attribut des produits que nous fabriquons ou utilisons. Cette qualité devient une arme comparative entre produits. Mais qu’est-ce que la qualité ? Pour [Kitchenham 96], « la qualité est difficile à définir, impossible à mesurer et facile à connaître ».

En Interaction Homme-Machine, on a toujours parlé de l’utilisabilité des systèmes interactifs. L’utilisabilité a été introduite pour essayer de comprendre quels facteurs contribuent à une interaction réussie entre l’Homme et la Machine. Pour concevoir des systèmes utilisables, il fallait avoir des critères mesurables à référencer tout au long du processus de développement.

Plusieurs référentiels ont été proposés, provenant aussi bien d’ergonomes que d’informaticiens. Les terminologies utilisées divergent : on parle de facteurs, critères et caractéristiques. Les décompositions diffèrent elles aussi. La figure 72 collecte quelques référentiels, sans pour autant les comparer. La juxtaposition montre

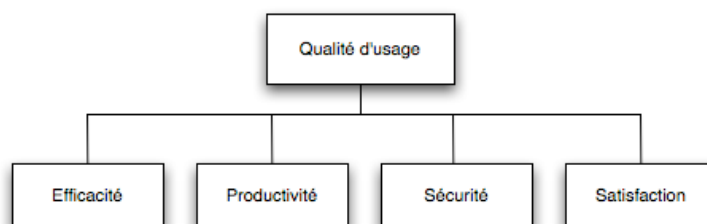
clairement l'absence de consensus. C'est l'apprentissage qui se retrouve dans le plus de travaux. Mais il n'est cité que trois fois sur cinq.

<b>Bastien et Scapin [Bastien 83]</b>	<b>Shackel [Shackel 91]</b>	<b>Nielsen [Nielsen 93]</b>	<b>IFIP [Gram 96]</b>	<b>ISO [ISO 03]</b>
<ul style="list-style-type: none"> <li>❶ Charge de travail</li> <li>❷ Guidage</li> <li>❸ Contrôle explicite</li> <li>❹ Adaptabilité</li> <li>❺ Gestion des erreurs</li> <li>❻ Homogénéité/ Cohérence</li> <li>❼ Signifiante des codes et dénominations</li> <li>❽ Compatibilité</li> </ul>	<ul style="list-style-type: none"> <li>❶ Efficacité</li> <li>❷ Apprentissage</li> <li>❸ Flexibilité</li> <li>❹ Attitude</li> </ul>	<ul style="list-style-type: none"> <li>❶ Apprentissage</li> <li>❷ Efficacité</li> <li>❸ Mémorabilité</li> <li>❹ Erreurs</li> <li>❺ Satisfaction</li> </ul>	<ul style="list-style-type: none"> <li>❶ Flexibilité</li> <li>❷ Robustesse</li> </ul>	<ul style="list-style-type: none"> <li>❶ Compréhensibilité</li> <li>❷ Apprentissage</li> <li>❸ Opérabilité</li> <li>❹ Attractivité</li> <li>❺ Complétude</li> </ul>

*Figure 72 : Quelques référentiels pour raisonner sur l'utilisabilité. Dans ces référentiels, certains critères ont été affinés. Ces affinements ne sont pas rapportés ici. C'est le cas typiquement de l'IFIP.*

Une particularité de l'ISO [ISO 03] est de distinguer trois types de qualité : la qualité interne, externe et à l'usage. Ce recul permet d'intégrer les perspectives IHM et Génie logiciel dans une vision unifiée de la Qualité.

La qualité à l'usage désigne la qualité du logiciel telle que perçue par l'utilisateur. Elle s'évalue en situation, dans un contexte d'usage donné. Elle se décline dans la norme en quatre caractéristiques (Figure 73) : l'efficacité, la productivité, la sécurité et la satisfaction.



*Figure 73 : La qualité à l'usage selon la norme ISO.*

La qualité à l'usage dépend des qualités externe et interne du logiciel. La qualité externe [ISO 03] désigne le degré selon lequel un produit répond aux besoins de l'utilisateur lorsqu'il est utilisé sous certaines conditions. La qualité interne porte sur

tous les attributs du produit qui contribuent à la satisfaction du besoin dans les conditions ciblées.

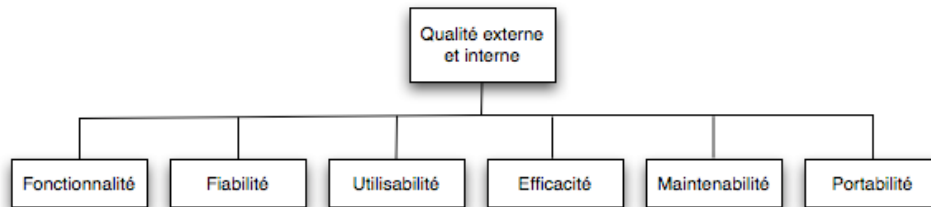


Figure 74 : Qualités externe et interne selon l'ISO.

Devant le manque de consensus dans la terminologie, [Seffah 01] propose un cadre intégrateur de la qualité d'usage. Ce cadre, appelé QUIM (Quality in Use Integrated Map) prend la forme d'une pyramide à plusieurs niveaux dont, en particulier, la qualité à l'usage, les facteurs, critères, métriques et données (Figure 75). L'idée est de localiser/ventiler au fil de ces niveaux les référentiels issus de l'IHM et du Génie logiciel. La relation entre niveaux est de type décomposition. Sa cardinalité est N-M. Ainsi, par exemple, un critère peut se décliner en plusieurs métriques mais une métrique peut servir plusieurs critères.

Selon QUIM, la qualité à l'usage est le point de vue utilisateur de la qualité du logiciel. Au niveau facteur, on trouve l'efficacité, l'efficience, la satisfaction, la productivité, la sécurité, l'internationalité et l'accessibilité. Comme critères, [Seffah 01] mentionne l'attractivité, la cohérence, les actions minimales, la charge mémoire minimale et la complétude. Ces critères rejoignent en partie [Bastien 93]. Deux métriques sont citées : la concordance de la tâche et la cohérence visuelle [Constantine 99]. Les mesures s'appuient sur des données mises en socle de la pyramide. Ces données peuvent être quantitatives ou qualitatives.

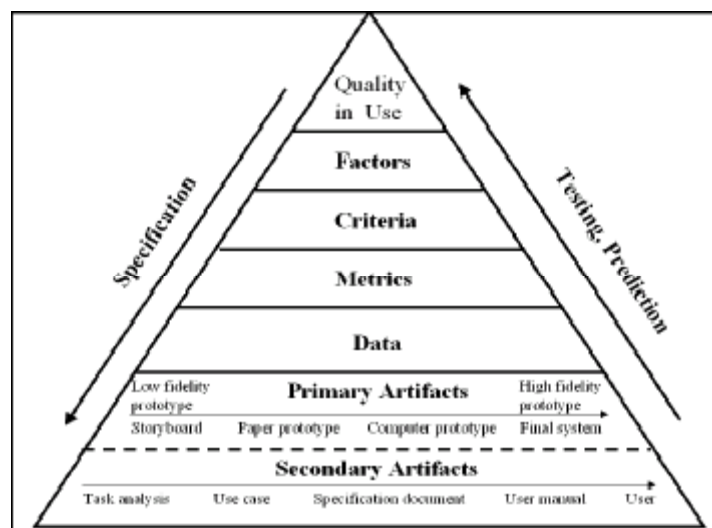


Figure 75 : Le cadre QUIM pour les modèles de qualité [Seffah 01].

QUIM peut servir à positionner et donc comparer les différents référentiels qu'ils soient issus du Génie logiciel ou de l'IHM. La section suivante répertorie les travaux spécifiques à l'adaptation.

### 3.2.2 Adaptation

Les deux perspectives fonctionnelle et non fonctionnelle sont traitées dans cet ordre.

#### Perspective fonctionnelle

[Dieterich 93] structure le processus d'adaptation en quatre étapes :

- L'*initiative* relative à la prise de décision quant à la nécessité d'adapter ;
- La *suggestion* de réactions ;
- La *décision* quant au choix de la réaction ;
- L'*exécution* relative à la mise en œuvre de la réaction.

[Calvary 01] réorganise ces étapes en trois étapes, chacune étant affinée en trois sous-étapes :

- La reconnaissance de la situation comprend :
  - La capture du contexte d'usage (ex : température de 22°) ;
  - La détection du changement (ex : passage de 21° à 22°) ;
  - L'identification du changement (ex : transition du contexte *Normal* au contexte *Confort*). C'est cette identification qui va déclencher une éventuelle réaction. [Schmidt 99] distingue trois types de déclencheurs : l'entrée dans un contexte ; la sortie d'un contexte et le fait d'être dans un contexte. Les déclencheurs sont combinables par conjonction et disjonction. Ainsi "sortie(C1) et entrée(C2)" est un déclencheur exprimant la transition du contexte C1 au contexte C2 ;
- Le calcul de la réaction couvre :
  - Le recensement des réactions possibles. [Calvary 01] prévoit :
    - Une intervention sur le contexte d'usage (ex : passer du portable au Palm lorsque la batterie faiblit ou allumer la lumière lorsque la pièce s'obscurcit) ;
    - Un changement d'exécutable. L'IHM courante ne couvrant pas le nouveau contexte, l'exécution d'une nouvelle IHM s'impose ;
    - Une adaptation de l'IHM par remodelage et/ou redistribution ;
    - L'exécution de tâches par le système et/ou l'utilisateur (ex : déclencher automatiquement le chauffage à l'approche de l'occupant) sans modifier la présentation. Seul le dialogue en est éventuellement modifié ;

- Le choix d'une solution parmi les candidates. Plusieurs critères peuvent être considérés, en particulier les ressources requises et propriétés garanties ;
- La mise en œuvre consiste à :
  - Préparer la réaction. Ce prologue comprend la gestion de la tâche en cours (terminaison, suspension ou avortement), la sauvegarde du contexte d'interaction (ex : température en cours de modification) et la production effective du changement (nouvelle présentation, nouveau dialogue) si cette nouvelle version n'a pas encore été élaborée ;
  - Exécuter la réaction (ex : commuter vers la nouvelle présentation, le nouveau dialogue ou exécuter une tâche) ;
  - Terminer la réaction. Cet épilogue comprend la restauration du contexte d'interaction sauvegardé lors du prologue.

Chaque étape peut être mise en œuvre par le système et/ou l'utilisateur, à la volée ou en différé. Dès lors qu'une étape se fera en différé, il en sera de même pour les suivantes. [Calvary 01] précise que l'observabilité de chaque étape sera à étudier conformément aux propriétés définies en IHM. Les propriétés générales ont fait l'objet de la section 3.2.1. La section suivante traite des propriétés spécifiques à l'adaptation.

### **Perspective non fonctionnelle**

A notre connaissance, les référentiels généraux de l'ergonomie n'ont pas été revisités pour intégrer la préoccupation de l'adaptation. Seul [Lopez-Jaquero 04] ajoute la continuité de l'interaction aux critères de C. Bastien et D. Scapin [Bastien 83].

En cinématographie, la continuité est importante. Le film est pris et coupé de façon à ce que le flux d'action ait une apparence souple et logique [Harris 91]. Un film est construit par concaténation de scènes prises à des moments différents alors qu'elles sont censées représenter une histoire continue dans le temps et dans l'espace. Il existe différents types de continuité : continuité du contenu, continuité du mouvement, continuité de la position, continuité du temps. Les cinématographes adoptent des techniques bien spécifiques pour maintenir la continuité.

En interaction Homme-Machine, différents types de continuité ont de même été proposés. Le projet européen TACIT (Theory and Application of Continuous Interaction Techniques) traitait spécifiquement de la continuité de l'interaction. Un cadre de référence (Figure 76) fut proposé [Massink 02]. Il est inspiré du protocole de communication ISO, tout en voulant intégrer des modèles issus de la psychologie cognitive tels ICS ou la Théorie de l'Action (section 2.2.1). Le cadre de référence modélise l'interaction à cinq niveaux : physique, perceptuel, propositionnel, conceptuel et groupe. Les trois niveaux perceptuel, propositionnel et conceptuel représentent le traitement de l'information. Le niveau groupe est relatif à la coordination des tâches en travail coopératif. Le niveau le plus bas est le niveau physique. C'est à ce niveau que l'utilisateur reçoit l'information de son

environnement. Cette information est reçue via les sens humains qui réagissent à des signaux physiques.

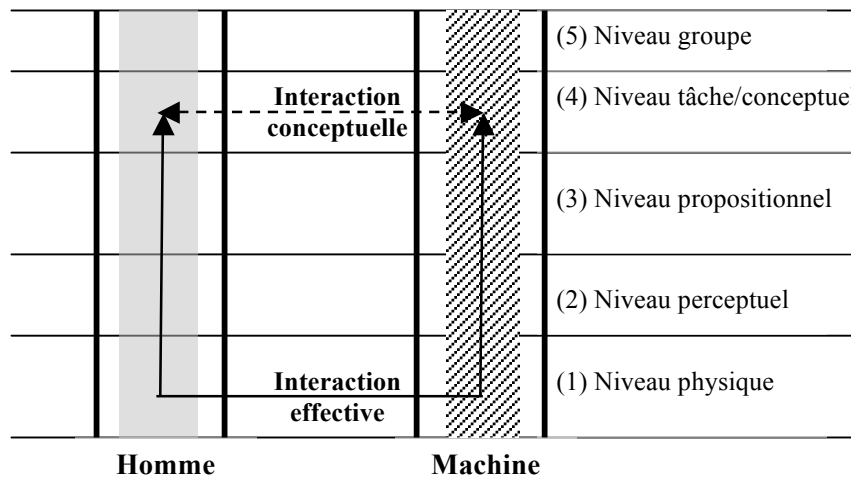


Figure 76 : Modèle de référence pour l'interaction Homme-Machine continue par [Massink 02].

La continuité de l'interaction fut récemment ré-examinée dans le cas particulier des systèmes mixtes pour lesquels les mondes physique et numérique cohabitent. [Dubois 01] [Trevisan 03] en sont des exemples. Récemment, [Florins 04] distingue trois niveaux de continuité :

- La continuité *perceptuelle* est définie comme la capacité du système à rendre observables toutes les données pertinentes du système vis-à-vis de la tâche utilisateur dans un même environnement perceptuel, afin d'éviter à l'utilisateur tout changement de focus ;
- La continuité cognitive est définie comme la capacité du système à garantir que l'utilisateur va interpréter correctement l'information perçue et que cette information perçue est conforme à l'état interne du système. On retrouve ici les propriétés d'observabilité et d'honnêteté de l'IFIP [Gram 96] ;
- La continuité fonctionnelle est relative à l'adaptabilité de l'utilisateur pour changer ou apprendre de nouveaux modes opératoires.

En pratique, à l'image des démonstrations par l'absurde en Mathématiques, la continuité est aussi étudiée via son dual : la discontinuité. Par exemple, [Tan 03] étudie l'effet de la discontinuité visuelle résultant de l'affichage de l'information sur plusieurs dispositifs d'affichage (Figure 77). L'étude montre que cette discontinuité physique n'a aucun effet sur la performance de l'utilisateur dans la réalisation de sa tâche.

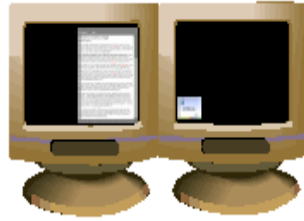


Figure 77 : *Discontinuité physique : cas de la distribution de systèmes.*

La section suivante dresse une synthèse de cette revue d'acquis pluridisciplinaires.



## 4. Synthèse

Ce chapitre montre la complexité du problème dans ses niveaux d'abstraction et les angles d'attaque possibles. Si d'un point de vue physique, le recensement des ressources d'interaction requises pour les entrées et les sorties semble possible aussi bien d'un point de vue humain que système, l'exercice au niveau cognitif est beaucoup plus périlleux. Il existe des modèles ou théories permettant de prédire ou expliquer la complexité de l'activité humaine (typiquement ICS, par le nombre de sous-systèmes mobilisés et la complexité des flux d'information). En revanche, il n'existe pas de *formule* permettant de la calculer. Aussi, pensons-nous raisonnable de :

- Recenser les ressources d'interaction physiques humaines et système requises ;
- Nous appuyer sur des référentiels existants pour palper la complexité de l'activité cognitive et système.

L'absence de consensus sur les modèles appelle à une tolérance dans le choix des référentiels : il semble judicieux de s'ouvrir à différents référentiels, moyennant, en conséquence, un effort d'alignement comme esquissé dans [Florins 04] qui met en vis-à-vis [Dubois 01] et [Gram 96]. Cette ouverture rejoint les travaux menés dans d'autres communautés telles que l'IDM qui clôt les éternels débats de comparaison, par exemple, de langages de programmation (C++ versus Java versus Tcl, etc.) par la notion de *passerelles* entre espaces technologiques : chaque espace a ses forces et faiblesses. Il s'agit donc, de façon pratique, de se placer dans l'espace technologique le plus approprié pour résoudre une classe de problèmes quitte, si nécessaire, à changer d'espace dès lors que ce dernier n'est plus indiqué pour raisonner sur une nouvelle classe de problèmes. De même, chaque référentiel (par exemple, IFIP, Bastien & Scapin, etc.) a ses forces et faiblesses. Il s'agit donc de permettre à chaque acteur en fonction de la classe de problèmes à résoudre et de son expertise dans les différents référentiels de choisir celui qui lui semble le plus approprié pour raisonner.

D'un point de vue terminologique, nous retenons de [Halford 93] la distinction entre *ressources*, *capacité* et *charge cognitive*, la charge cognitive étant appelée *coût* par [Barouillet 96]. Nous généralisons ces termes pour les aspects autres que cognitifs et appellerons :

- *Ressources* : un capital disponible dans une entité donnée, à un instant donné pour une classe particulière de traitements ;
- *Capacité* : la quantité maximale de ressources mobilisables à un instant donné par une entité donnée ;
- *Coût* : la quantité de ressources mobilisées dans une entité donnée pour la réalisation d'une tâche donnée.

En Marketing, [Broekhuizen 03] distingue deux types de coûts :

- Un coût monétaire qui dénote le prix du produit ;
- Un coût non monétaire qui couvre le temps, l'effort et l'effet psychologique que le produit aura sur l'utilisateur.

Dans l'esprit de QUIM, nous généralisons ces travaux en distinguant trois facteurs de coûts :

- La satisfaction des pré-conditions : elle va de l'acquisition ou mobilisation des entités requises pour l'interaction jusqu'à la satisfaction des contraintes (par exemple, l'utilisateur fait face à l'écran) ;
- L'exploitation des entités dans l'interaction. D'un point de vue système, l'exploitation couvre les opérations d'entrée, sortie, calcul et communication. D'un point de vue utilisateur, l'exploitation couvre les cycles d'exécution et évaluation tels qu'exprimés dans la Théorie de l'Action de Norman par exemple. Il est à noter qu'une ressource peut être utilisée de façon exclusive ou, au contraire, communautaire. Typiquement, les fenêtres transparentes se partagent une même surface d'affichage : on parlera de coût *commun* versus *propre*. Les travaux en multimodalité pourraient ici être pertinents. On se souvient des usages alterné, synergique, exclusif et concurrent de [Nigay 94] ensuite affinés dans [Vernier 01]. Les référentiels d'utilisabilité tels que [Bastien 93] fournissent aussi ici un ensemble de critères applicables ;
- Les effets post-exploitation : typiquement, la fatigue physique, les changements sonores, lumineux, etc. de l'environnement entrent dans cette esкарelle.

A noter que dès lors qu'une ressource sera partagée, la satisfaction des pré-conditions et l'exploitation se feront de façon itérative.

Nous identifions cinq critères selon la matérialité de la ressource consommée :

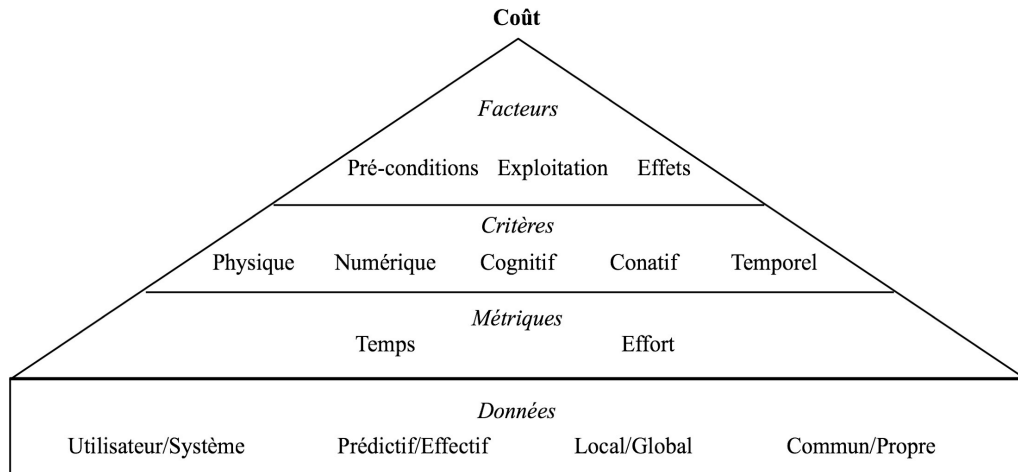
- Physique : il couvre, en particulier, l'effort physique de manipulation des entités physiques pour la satisfaction des pré-conditions ou l'exploitation ;
- Numérique d'un point de vue système ;



- Cognitif d'un point de vue humain pour la réalisation d'un cycle d'exécution-évaluation pour un but donné ;
- Conatif d'un point de vue humain pour la formulation d'un but ;
- Temporel quelle que soit la perspective système ou utilisateur.

Si la littérature abonde de critères, les métriques se font, en revanche, rares. Typiquement, le référentiel de C. Bastien et D. Scapin énonce huit critères sans que des métriques ne leur soient attachées. D'un point de vue expérimental, le temps et l'effort sont des métriques classiques. A noter que, quelle que soit la métrique, on peut évaluer le coût de façon prédictive ou effective, locale ou globale dans le temps (donnant alors des mesures instantanées ou cumulées), les fonctions, etc., se limiter aux ressources exclusives ou l'étendre aux ressources partagées.

La Figure 78 résume notre espace problème des coûts.



*Figure 78 : Espace problème des coûts.*

Dans ce chapitre nous avons essayé de combler les manques de l'état de l'art en matière d'interacteur et surtout en matière de valeur. Le chapitre était un parcours en largeur des disciplines en IHM et hors IHM (psychologie cognitive par exemple). Ce parcours, quoique pas très approfondie, nous a permis d'établir un espace problème des coûts.

Le chapitre suivant présente notre modélisation des interacteurs pour la plasticité des interfaces Homme-Machine : les COMETS.

# CHAPITRE IV

## LES COMETS

---

Face aux manques de l'état de l'art, ce chapitre introduit une nouvelle génération d'interacteurs, les COMETS (COntext Mouldable widgETs), pour la plasticité des systèmes interactifs. Une définition puis une classification en sont données dans les deux premières sections. Les deux suivantes concernent la modélisation et l'adaptation de la COMET. Un modèle d'architecture logicielle est ensuite proposé. Le chapitre se termine par une synthèse montrant comment ce nouveau concept de COMET comble l'état de l'art et répond au problème de plasticité à la granularité des interacteurs.

---

### 1. Définition

Une COMET (COntext Mouldable widgET) est un interacteur doué d'autodescription fonctionnelle, ontologique, génétique et de valeur.

Comme pour un interacteur, la granularité d'une COMET reste à l'appréciation du concepteur. Elle peut aller de l'interacteur classique (dit d'utilité publique) comme le champ texte, le bouton, etc., au système interactif spécifique tel que PlasticClock par exemple (Figure 19). L'intérêt de déclarer un système interactif en tant que COMET est la réutilisation qui en résulte. Si typiquement, la double horloge est une fonctionnalité classique dans le domaine applicatif traité, alors il peut être intéressant de la capitaliser comme COMET réutilisable.

La section suivante propose une classification de COMETS.

---

## 2. Classification

Contrairement à la littérature qui, de façon récurrente, propose de nouvelles classifications fonctionnelles métier, nous proposons ici deux classifications originales : la première relative à l'autodescription de la COMET ; la seconde relative à son potentiel d'adaptation. On retrouve ainsi les deux parties « Autodescription » et « Potentiel d'adaptation » de l'espace problème.

### 2.1 Autodescription

Nous identifions trois critères discriminants en matière d'autodescription :

- La précision de la description : la description est-elle une description de genre, c'est-à-dire d'espèce, ou, au contraire, d'individu ? Typiquement, un champ texte dira-t-il « J'offre la tâche Spécifier concept » ou « Je suis la matérialisation de *cette* tâche Spécifier *un certain* concept dans le modèle des tâches » ?
- Le moment d'élaboration de la description : la description est-elle préfabriquée à la conception ou calculée à l'exécution ?
- Le support de la description : la description est-elle implicite, c'est-à-dire certes portée par l'objet, mais non exprimée en tant que telle, ou est-elle explicite portée par l'objet ou un méta-objet (on la dira alors respectivement interne et externe) ?

L'autodescription peut s'appuyer sur des mécanismes d'introspection ou de réflexion [Englander 01].

La Figure 79 résume cette classification.

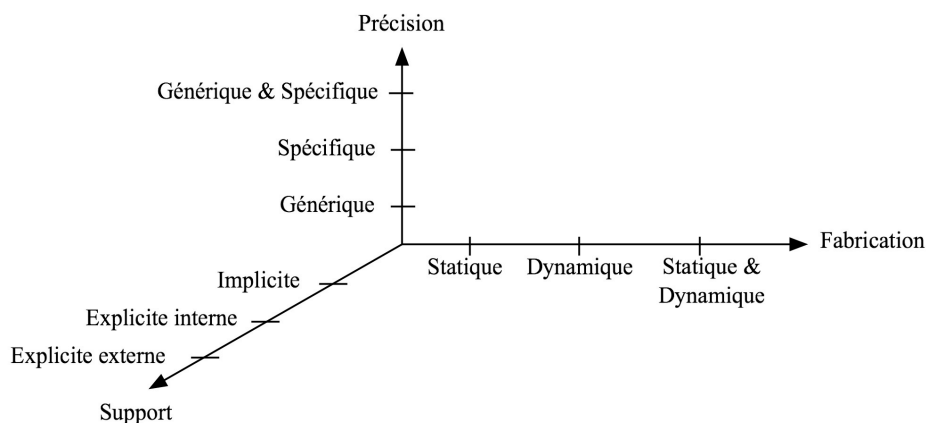


Figure 79 : Classification des COMETs en matière de description.

## 2.2 Adaptation

En matière d'adaptation, nous identifions trois gammes de COMETs :

- Les COMETs basiques, c'est-à-dire 'simplement' autodéscriptives tel qu'imposé dans la définition ;
- Les COMETs polymorphes, c'est-à-dire à présentations et/ou abstractions multiples ;
- Les COMETs autoadaptatives, c'est-à-dire douées d'adaptation.

Dans leur forme la plus basique, les COMETs sont autodéscriptives, rendant inspectables leurs caractéristiques fonctionnelles, ontologiques, génétiques et de valeur telles que définies dans l'espace problème.

Dans la lignée de M. Crease [Crease 01], les COMETs peuvent être polymorphes. Elles restent, bien entendu, autodéscriptives mais dotées d'un ensemble d'incarnations en présentation et/ou abstraction. Ce potentiel peut être précieux en cas de changement de contexte d'usage, les incarnations pouvant se distinguer dans les ressources qu'elles requièrent ou les propriétés qu'elles garantissent. Le polymorphisme peut s'exercer à différents niveaux d'abstraction : concepts et tâches, interfaces abstraites, concrètes et finales. PlasticClock, par exemple, est polymorphe : elle embarque les présentations de la Figure 19 (a, b et c). Le polymorphisme s'y exerce aux niveaux tâche (jouant sur la disponibilité de la tâche « consulter date » seulement offerte en version Figure 19d), interfaces abstraites (par le partage ou non d'un même espace de travail pour représenter les deux heures), concrètes (par le choix de rendu texte/cadran) et finales (par l'usage de boîtes à outils spécifiques pour l'exécution sur PC ou téléphone). En cas de changement de contexte d'usage, PlasticClock, dans son adaptation interne, joue sur la présentation qu'elle choisit d'afficher.

Les COMETs auto-adaptatives sont les COMETs les plus riches fonctionnellement. Elles embarquent, en plus de leur capacité métier, des mécanismes d'adaptation (reconnaissance du contexte d'usage, calcul et/ou mise en oeuvre de la réaction). La réaction peut mettre à profit l'éventuel polymorphisme de la COMET.

La Figure 80 résume cette classification des COMETs. Elle précise pour chaque type (autodéscriptif, polymorphe, auto-adaptatif), la localisation de l'adaptation : externe pour une COMET basique et/ou polymorphe, potentiellement interne pour une COMET auto-adaptative mais l'adaptation peut y être externe et/ou mixte.

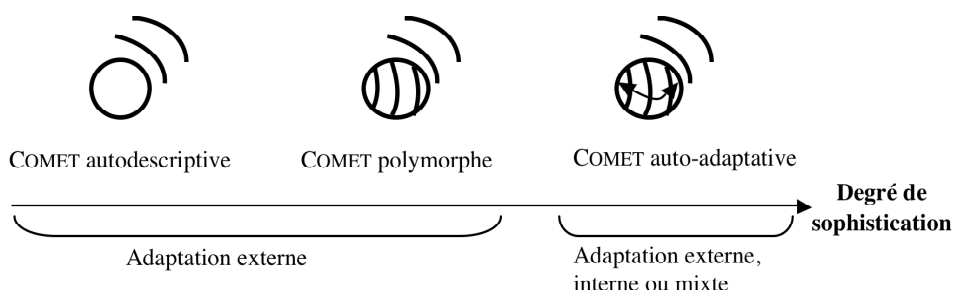


Figure 80 : Classification des COMETs en matière de description.

Il est intéressant de noter que, pour des COMETs dont le métier est l'adaptation, ces fonctions d'adaptation relèvent du niveau métier. Ce sont alors des COMETs de l'ordre de la méta-IHM : elles offrent des tâches utilisateur telles que *Remodeler*, *Migrer*, etc. Elles peuvent elles-mêmes embarquer des fonctions d'adaptation : la méta-IHM devient alors plastique. Cette réflexion nous conforte dans la difficulté d'établir une classification fonctionnelle. Les travaux sur les ontologies et alignements d'ontologies pourraient ici apporter une réponse.



### 3. Modélisation

La modélisation s'appuie sur le cadre de référence révisé de la plasticité (Chapitre III, Section 1) : une COMET est un interacteur dont la description est multifacettes, les facettes relevant potentiellement de quatre niveaux d'abstraction :

- Tâches-Concepts, ici appelé Abstraction ;
- Interface abstraite ;
- Interface concrète ;
- Interface finale.

Les niveaux se correspondent par transformations. Chaque niveau et chaque transformation décrivent leur valeur en termes des ressources qu'ils requièrent et des propriétés qu'ils garantissent. Seul le niveau Abstraction est requis. Une COMET polymorphe est une COMET dont la cardinalité des Interfaces concrètes est supérieure ou égale à deux.

Une facette Adaptation est requise pour la description des éventuelles capacités d'adaptation de la COMET. Que la COMET soit auto-adaptative ou non, elle comporte cette facette d'Adaptation.

La Figure 81 est un modèle UML simplifié de COMET. Une version plus élaborée en est donnée en synthèse. La contrainte OCL imposant l'existence d'au moins une abstraction et d'une adaptation n'est pas fournie.

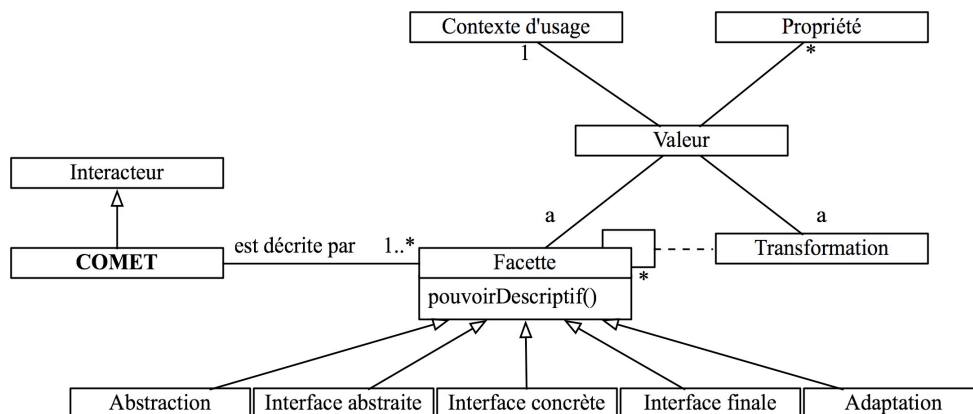


Figure 81 : Modèle simplifié de COMET.

Cette section examine dans un premier temps la description multifacettes puis la façon selon laquelle propriétés et ressources s’y ventilent.

### 3.1 Description multifacettes

Jusqu’ici les niveaux d’abstraction étaient considérés comme des étapes dans le processus de construction d’une IHM. Ils sont désormais perçus comme des perspectives différentes sur un même objet d’étude qu’est l’interacteur. L’interacteur se décrit à différents niveaux d’abstraction allant du niveau Tâches-Concepts à l’interface finale. En particulier, l’interface finale est une description de l’interacteur selon une perspective logique.

Nous rappelons ci-dessous la portée de chaque niveau.

#### ☛ Abstraction

L’abstraction de la COMET décrit la tâche utilisateur qu’offre la COMET. La tâche est classiquement modélisée par un arbre des tâches. L’arbre décompose récursivement la tâche en sous-tâches jusqu’à atteindre un niveau dit « tâches élémentaires » indépendant des dispositifs d’entrée et de sortie. Les tâches intermédiaires sont dites « abstraites ». Les tâches sont reliées entre elles par des opérateurs logiques et/ou temporels. Des formalismes tels que ConcurTaskTree [Paterno 97] ont été proposés. Ce dernier reprend les opérateurs Lotos.

Les tâches sont décorées, en particulier, des concepts du domaine qu’elles manipulent. D’autres décorations sont possibles. On trouve classiquement la fréquence, l’optionnalité, l’itérativité, la criticité ou la complexité. [Brisson 94] permet, dans PROSPECT, la définition de pré- et post-conditions en remplacement de certains opérateurs.

Prenons deux exemples de modèles de tâches pour illustrer les niveaux et leur réification :

- Une séquence entre sous-tâches (Figure 82a) : T s'affine en T1 puis T2 ;
- Un entrelacement entre sous-tâches : T est réalisée lorsque ses filles T1, T2 et T3 le sont, la réalisation des filles étant entrelacée (Figure 82b). T1 s'affine, de même, en un entrelacement de quatre sous-tâches.

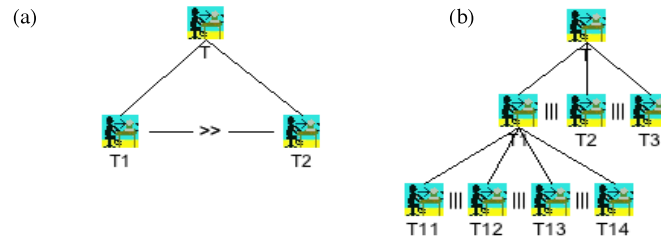


Figure 82 : Deux exemples d'abstraction édités dans l'environnement CTTE.

Si la description est générique (cf le pouvoir descriptif de la COMET), l'arbre des tâches est un arbre prototypique. Si la description est, au contraire, spécifique, alors la description *représente* ou *pointe sur* la partie du modèle de tâches que couvre précisément la COMET (*représente* ou *pointe sur* selon que le contrôleur de dialogue gère ou non un modèle des tâches global au système interactif).

### ☛ Interface abstraite

L'interface abstraite structure l'IHM en espaces de travail. Nous verrons, dans la section suivante, que la notion d'espace de travail est motivée par l'ergonomie qui recommande le groupement géographique d'entités (concepts et/ou tâches) sémantiquement liées. Les espaces de travail se donnent accès mutuellement. Si la compatibilité par rapport au modèle des tâches est garantie, alors l'enchaînement entre espaces est conforme aux opérateurs qui lient les tâches auxquels ces espaces correspondent.

Reprenons la séquence entre tâches (Figure 82a). L'interface abstraite (Figure 83) identifie trois espaces de travail : un pour la tâche mère T ; un pour chacune des deux filles T1 et T2. L'espace de travail de T donne accès à l'espace de travail de T1 qui donne accès à l'espace de travail de T2.



Figure 83 : Un exemple d'interface abstraite pour le modèle des tâches donné en Figure 82a.

De même, pour l'entrelacement, une heuristique peut être d'accorder un espace de travail par tâche utilisateur. La Figure 84 donne un exemple d'interface abstraite.

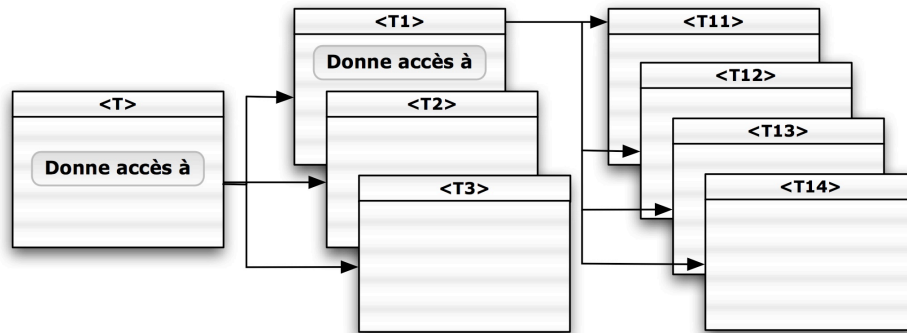


Figure 84 : Un exemple d'interface abstraite pour le modèle des tâches donné en Figure 82b.

Les descriptions peuvent, de même, être génériques et/ou spécifiques.

### ☛ Interface concrète

L'interface concrète est une matérialisation de la COMET en widgets. Les widgets y sont logiques, le choix physique de la boîte à outils n'ayant pas encore été fait.

Chaque élément du modèle des tâches et de l'interface abstraite doit normalement trouver corps dans l'interface concrète :

- Les fenêtres, régions, boîtes de groupement, etc., sont la matérialisation d'espaces de travail associés aux tâches abstraites. Les libellés, champs texte, images, etc., correspondent aux tâches élémentaires et concepts du domaine. Leur fonction sera de service ou d'estime. Un logo typiquement rendra une fonction d'estime pour une tâche donnée ;
- Les séparateurs, menus, onglets, liens hypertexte, boutons de navigation, etc., sont la matérialisation des opérateurs entre tâches : « Donne accès à ». Les figures 85 et 86 proposent, par exemple, deux interfaces concrètes pour les Figures 82a et 82b : les différences sont de l'ordre de la navigation.



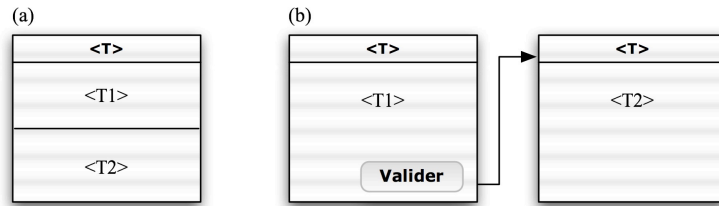


Figure 85 : Deux esquisses d'interfaces concrètes pour la Figure 82a. L'opérateur de séquence est matérialisé en a) par le séparateur et le grisé ; en b) par le bouton « Valider ».

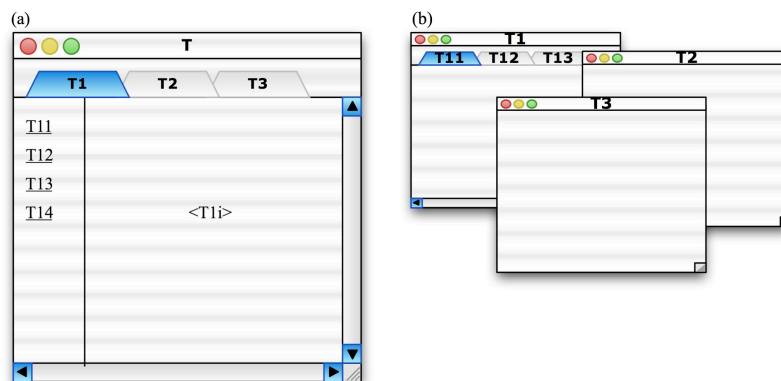


Figure 86 : Deux esquisses d'interfaces concrètes pour la Figure 82b. L'entrelacement est représenté en a) par des onglets et des liens hypertexte ; en b) par un multi-fenêtrage et des onglets.

- Les décorations entre tâches prennent aussi corps dans l'interface concrète. Par exemple :
  - L'itération peut justifier l'introduction d'un nouveau concept (par exemple, le panier) rendant observable le nombre d'itérations. Un bouton « Nouvelle commande » donnera alors accès à une nouvelle itération ;
  - La criticité peut se répercuter par une demande de confirmation ou le recours à des mises en évidence du type couleur, soulignement, gras, etc. ;
  - La fréquence peut avoir une incidence sur les emplacements respectifs des différents espaces de travail (la tâche fréquente sera classiquement au centre de l'écran) voire une navigation nulle pour cette tâche contrairement aux autres mises en accès indirect (via un menu, par exemple) ;
  - En web, l'optionalité est classiquement indiquée par un astérisque ;
  - La complexité exigera un meilleur guidage ;
  - Etc.

Nous verrons, dans la section suivante, comment les critères d'ergonomie soutiennent, en réalité, ces différentes transformations. Mais retenons que chaque

élément de l'IHM a une raison d'être au risque sinon de transgresser le critère de Charge de travail , sous-critère Brièveté tel qu'exprimé dans le référentiel de C. Bastien et D. Scapin.

### ☛ Interface finale

L'interface finale relève de la description de l'implémentation de la COMET dans un langage de programmation et une boîte à outils donnés. Ce niveau décrit typiquement le modèle d'architecture logicielle auquel le code est éventuellement conforme.

En pratique, toutes ces transformations sont motivées par un compromis entre propriétés et ressources. Selon les arbitrages faits entre propriétés et ressources, les facettes peuvent être multiples. On couvre ainsi le polymorphisme à tout niveau d'abstraction, y compris Tâches-Concepts, ce qui rejoint les tâches polymorphiques de [Savidis 01] (Figure 87).

Les sections suivantes traitent respectivement des propriétés et ressources dans les transformations.

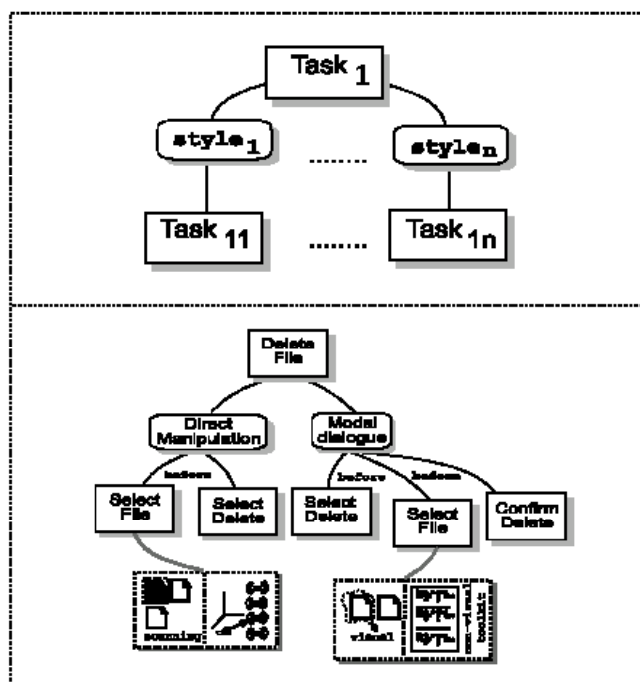


Figure 87 : Tâches polymorphiques de [Savidis 01].

## 3.2 Transformations et propriétés

A notre connaissance, les propriétés motivant les transformations dans le processus de construction d'IHM n'ont pas été explicitées. Nous nous plaçons dans le référentiel de C. Bastien et D. Scapin et montrons comment les critères s'y ventilent. La Figure 88 en donne une cartographie complète :

❶ Le modèle de tâches est motivé par le critère de *Compatibilité* par rapport à la tâche. Les transformations préserveront ensuite ou non ce modèle des tâches ;

❷ La transformation Tâches – Interface abstraite vise à garantir un groupement par la localisation des concepts et tâches sémantiquement connectées. C'est le critère de *Guidage*, sous-critère *Groupement/Distinction entre items par la localisation* qui est ici invoqué ;

❸ La transformation Interface abstraite-Interface concrète fait un choix de présentation pour les espaces de travail et leur contenu ainsi que les enchaînements entre espaces. Ces transformations sont dirigées par plusieurs critères dont la *Charge de travail*, sous-critère *Brièveté* : il s'agit d'afficher les concepts de façon concise et réduire la longueur de trajectoire d'interaction, en limitant, en particulier, les actions de type navigation qui ne contribuent pas à la réalisation de la tâche métier. Mais une trop forte *densité informationnelle* (deuxième sous-critère de la *Charge de travail*) peut inciter à l'introduction de navigations. Ce dosage est représentatif des critères à concilier à ce niveau. Interviennent aussi la *Signifiante des codes et dénominations* ; la *lisibilité*, l'*incitation* et le *retour d'information* (sous-critères du *Guidage*) ;

❹ L'*Homogénéité-Cohérence* est un critère s'assurant qu'à tout problème récurrent, une même solution est appliquée. Ce critère porte donc sur l'ensemble des transformations oeuvrant entre deux niveaux d'abstraction donnés : ces transformations doivent appliquer un même patron.

Restent trois critères :

- L'*Adaptabilité* peut être considérée dès le modèle de tâches si des tâches spécifiques de personnalisation par exemple sont intégrées au regard de ce critère. Si sa répercussion se limite, en revanche, à des raccourcis clavier, alors sa seule manifestation dans l'IHM est l'affichage des raccourcis dans les menus : ceci relève de l'interface concrète. En intermédiaire, nous avons les boutons de raccourcis qui justifient par contre un espace de travail dédié, regroupant ces raccourcis : ceci relève de l'interface abstraite ;
- La Gestion des erreurs a de même un ancrage variable dans le processus : elle peut être considérée dès le modèle des tâches prévoyant les situations anormales. Elle peut intervenir dans l'interface abstraite si des zones sont réservées à l'affichage des messages d'erreur. Elle peut aussi n'intervenir qu'au niveau de l'interface concrète par une mise en évidence du champ erroné ;
- Il en va de même pour le Contrôle explicite qui peut aller d'un contrôle fin de l'interaction à un simple 'Annuler' dont la trace se limite alors à l'interface concrète.

Enfin, la compatibilité par rapport à l'utilisateur et l'environnement relève des ressources. La section suivante y est consacrée.

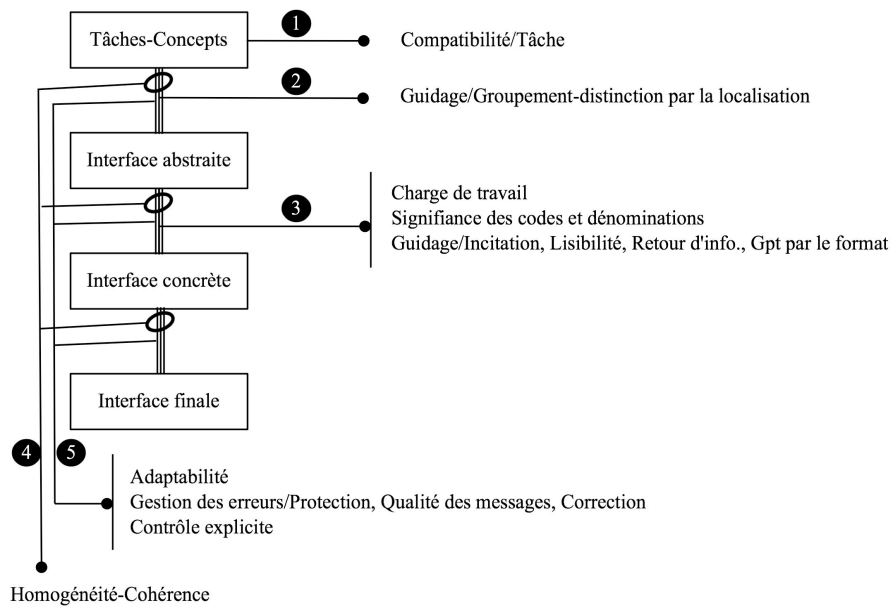


Figure 88 : Ventilation des critères de C. Bastien et D. Scapin dans le processus de construction d'IHM.

Comme discuté au chapitre III, il existe d'autres référentiels que C. Bastien et D. Scapin. L'analyse nous permet, par contre, de pointer que :

- Une transformation est motivée par des propriétés. Favoriser une propriété peut en défavoriser d'autres ;
- Une propriété peut porter sur un ensemble de transformations.

La Figure 89 est une description enrichie des COMETs. Une COMET est un interacteur. Elle est décrite par plusieurs facettes : abstraction, interface abstraite, interface concrète, interface finale et une facette adaptation. Les facettes se correspondent par transformation. Chaque facette décrit sa valeur en terme de contexte d'usage et de propriétés. Les propriétés sont décrites dans un référentiel tel que ISO.

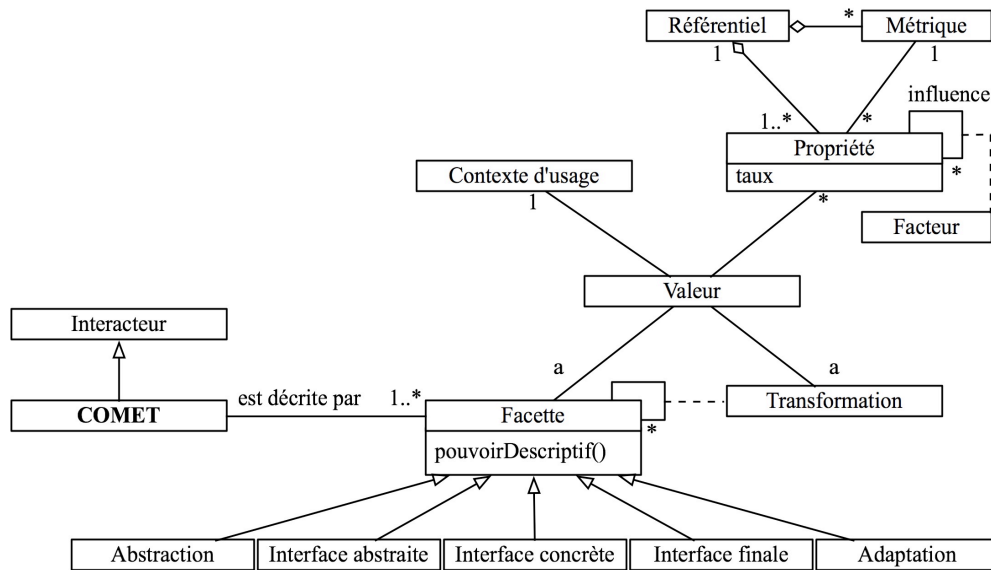


Figure 89 : Modèle enrichi de COMET.

### 3.3 Transformations et ressources

La construction d'une IHM est rythmée par un ensemble de compromis jouant sur les propriétés et ressources. Dans [Calvary 04], on identifiait quatre types de dépendance qu'on égrainait de façon verticale et descendante dans le processus de réification (Figure 90) :

- Le contexte d'usage ;
- La modalité ;
- L'interacteur ;
- L'environnement d'exécution.

Ces dépendances se comportaient comme des taquets mutuellement contraints. Aux extrémités, se trouvaient le contexte d'usage et l'environnement d'exécution. Ils bornaient ainsi la modalité et l'interacteur. Plus les dépendances étaient tardives (c'est-à-dire chassées dans les niveaux d'abstraction les plus bas), plus la production était générale, c'est-à-dire applicable et donc réutilisable.

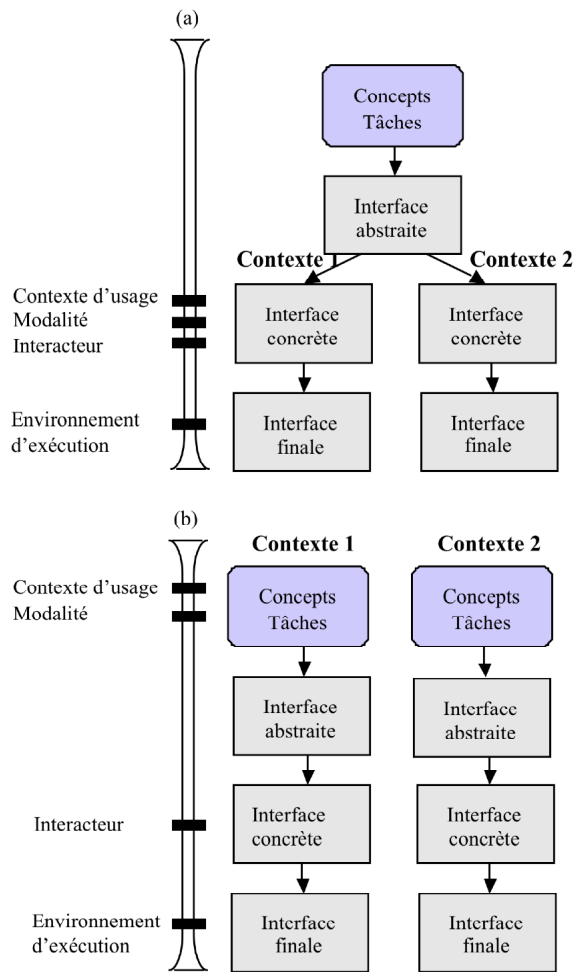


Figure 90 : Quatre types de dépendances mutuellement contraints, égrainés au fil du processus de construction.

Si les « pastilles » Interacteur et Environnement d'exécution relèvent clairement respectivement des niveaux Interfaces concrète et finale, nous sommes, en revanche, moins certains quant au positionnement des dépendances vis-à-vis du contexte d'usage et de la modalité. De notre point de vue :

- Certaines tâches ne font, par nature, pas sens dans certains contextes d'usage. En conséquence, les tâches deviennent dépendantes du contexte d'usage ;
- Le modèle des tâches peut, par ailleurs, être dépendant de la modalité (au sens humain). L'exemple typique est celui des modalités graphique et sonore pour lesquelles l'opérateur entre tâches varie. En graphique, la destruction d'un objet se fait classiquement par sélection de l'objet puis application de la commande « Détruire ». En vocal, la commande est, en revanche, spécifiée avant que l'objet ne soit sélectionné (« Détruis cet objet »).

En conséquence, il nous semble plus prudent de percevoir le contexte d'usage et la modalité comme des degrés de liberté que le concepteur fixe en fonction des compromis à tenir entre ressources et propriétés. Les facettes racontent si elles en sont ou non dépendantes.

Conformément à l'espace problème des coûts établi au chapitre III, nous examinons ci-dessous le coût de la COMET en termes de ressources physiques, numériques, cognitives, conatives et temporelles. Le coût est ici commun, faisant abstraction d'un éventuel partage de ressources avec une autre COMET ou interacteur. Par exemple, dans la Figure 91 extraite du démonstrateur (Chapitre V), la lentille magique est placée sur la liste scrollée. Les deux COMETs lentille et liste se partagent, en conséquence, une même surface d'affichage.

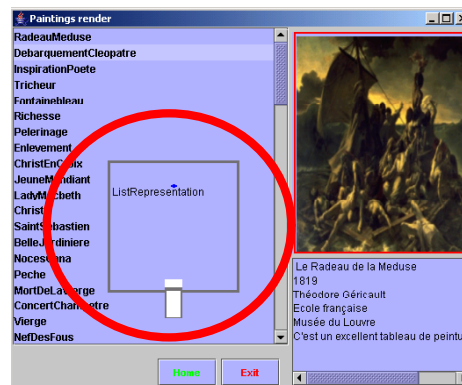


Figure 91 : La lentille magique est placée sur la liste scrollée : les deux COMETs se partagent une même surface d'affichage.

### 3.3.1 Ressources physiques

La plate-forme et l'utilisateur sont sources potentielles d'entités physiques. Si un rideau de pluie était utilisé comme surface d'affichage, alors nous considèrerions la pluie comme élément de la plate-forme et non de l'environnement.

#### ☛ Plate-forme

La plate-forme fournit les dispositifs système d'interaction : les capteurs et effecteurs qui soutiennent respectivement les observations et actions système. C'est le couplage des capteurs et effecteurs à des entités physiques qui les rend opérationnels (Figure 12). Et c'est le couplage des capteurs et effecteurs humains à ces mêmes entités physiques qui assure l'interaction Homme-Machine. Les sens humains sont alors mobilisés. Aussi s'agit-il de décrire pour chaque capteur et effecteur système :

- les caractéristiques des entités physiques les rendant éligibles au couplage ;
- la modalité humaine impliquée.

Notre ambition ici n'est pas l'exhaustivité. Il s'agit simplement d'esquisser le principe.

Pour l'observation, nous identifions trois types de capteurs (Figure 92) :

- Les capteurs de pointage tels que la souris, la boule roulante, la manette, etc. [Hinckley 03] ;
- Les capteurs de sons, d'images tels que les microphones ou caméras ;
- Les capteurs de spécification de données tels que les claviers, haut-parleurs, etc.

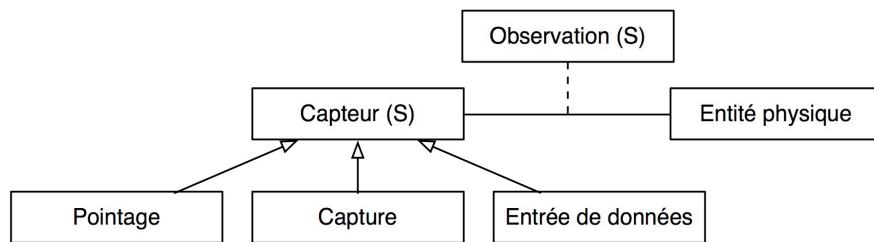


Figure 92 : Capteurs système.

Pour l'action, les cinq sens humains sont candidats. L'exposition peut être auditive, visuelle, olfactive, gustative et/ou tactile (Figure 93).

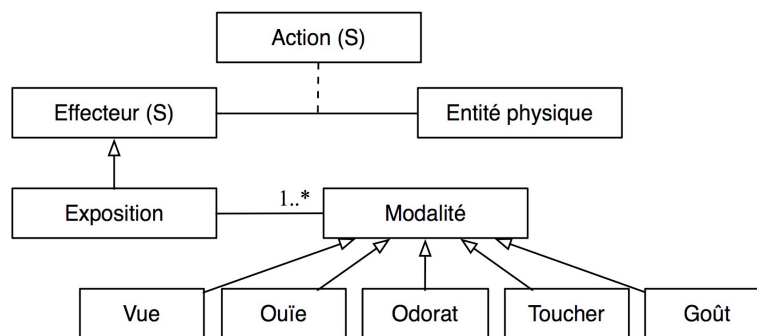


Figure 93 : Effecteurs système.

Le couplage peut se faire à la conception ou à l'exécution. Un couplage à l'exécution suppose une découverte des entités physiques et la possibilité de leur affecter un rôle d'interaction selon leurs caractéristiques et les requis. A noter que l'écran n'est autre qu'un couplage packagé de vidéo projecteur et d'une surface d'affichage. La surface d'affichage pourrait être la main, une table, un mur ou n'importe quelle surface d'entité physique dès lors que les caractéristiques de cette surface sont compatibles avec la scène à afficher.

## ☛ Utilisateur

Les capacités perceptuelles et motrices de l'utilisateur (respectivement Observation (H) et Action (H)) sont ici concernées.



Pour l'observation, les capteurs sont les yeux, la peau, les oreilles, le nez et la langue donnant lieu aux modalités visuelle, tactile, auditive, olfactive et gustative (Figure 94).

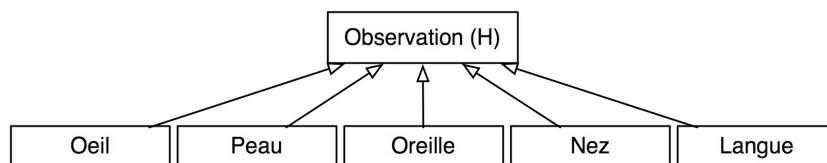


Figure 94 : Capteurs humains.

Pour l'action, les effecteurs sont classiquement la main. Mais les yeux, la tête et le corps peuvent être imaginés comme dispositifs d'entrée (Figure 95).

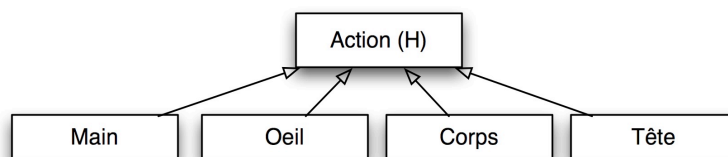


Figure 95 : Effecteurs humains.

### 3.3.2 Ressources numériques

Les ressources numériques sont celles nécessaires à l'exécution de la COMET. Nous citons par exemple la taille de la mémoire ou la vitesse du processeur. Les travaux en complexité permettraient de chiffrer les requis a priori. Ces travaux peuvent par exemple estimer, en fonction de la complexité des algorithmes, la taille mémoire nécessaire ou le temps d'occupation du processeur pour l'exécution d'un code. Le système par contre, connaît la consommation effective de ces ressources numériques. L'expression peut s'appuyer sur CC/PP [W3C 06].

### 3.3.3 Ressources cognitives, conatives et temporelles

Comme montré au chapitre III, le plan cognitif est délicat. En effet, il n'existe pas de métriques pour mesurer l'effort cognitif de réalisation d'une tâche. Le temps de réalisation est très souvent interprété comme un signe de cet effort. ICS par exemple, nous montre une décomposition du système cognitif humain sans pourtant fournir de méthodes et de métriques d'estimation de l'effort mental. Les tâches sont exprimées dans ICS en terme de communications entre les différents sous-systèmes intervenant dans la réalisation de la tâche en question. Nous pourrions utiliser la longueur du chemin liant les différents sous-systèmes comme métrique de mesure.

Les ressources conatives désignent le degré de plaisir apporté à l'utilisateur. L'évaluation de ces ressources se fait à travers les questionnaires avec les utilisateurs.

### 3.4 Synthèse

La Figure 96 fournit en synthèse la modélisation d'une COMET. Une réflexion en profondeur serait maintenant requise pour affiner chaque classe. Un ralliement vers des langages de description tels que UsiXML (<http://www.usixml.org/>) serait pertinent.

La COMET raconte si elle est polymorphe, autoadaptive ou plastique. Chaque contexte d'usage a un coût exprimé en termes de préconditions, exploitation et effets. Ce coût est relatif à des ressources physiques (utilisateur et plate-forme), numériques de la plate-forme, cognitives de l'utilisateur, conatives de l'utilisateur et temporelles.

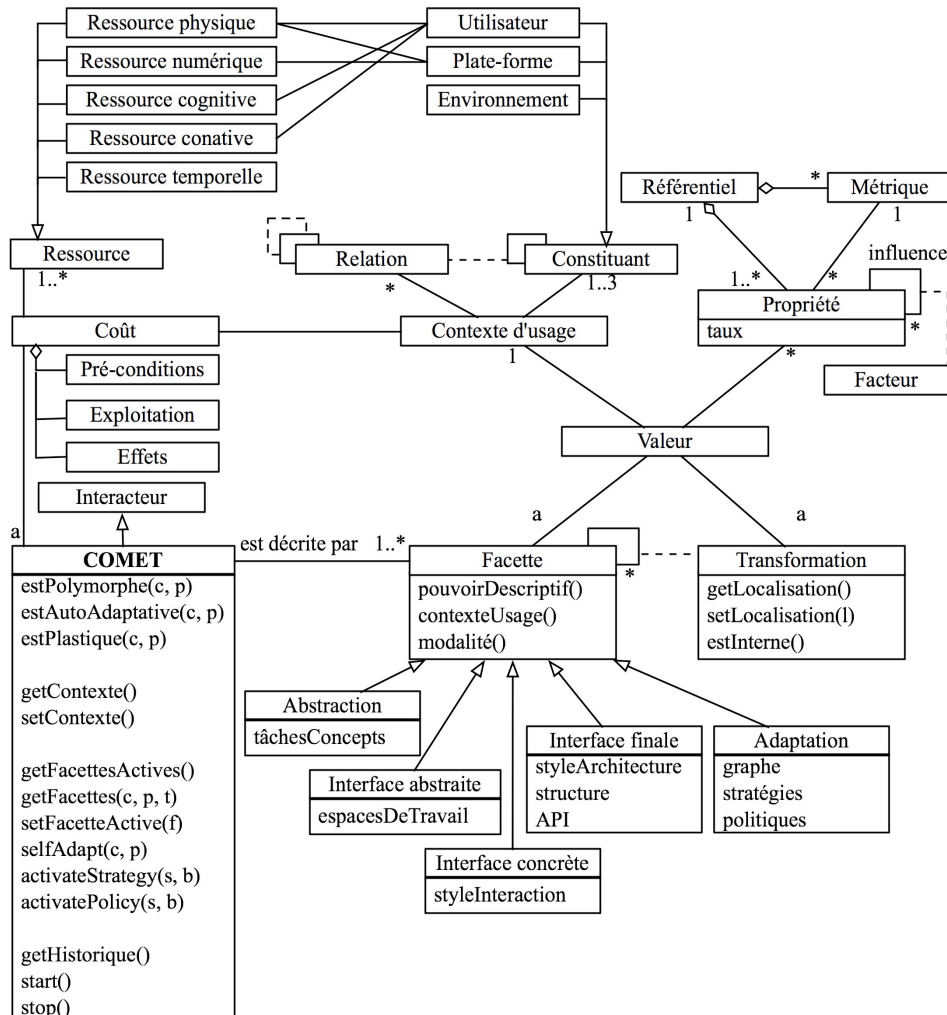


Figure 96 : Modèle de COMET.

La section suivante traite de l'adaptation.



## 4. Adaptation

Nous examinons dans cette section les COMETs sous l'angle de l'exécution. Nous définissons des stratégies et politiques d'adaptation.

### 4.1 Stratégies

Nous identifions deux gammes de stratégies : les stratégies de constitution et stratégies de valeur.

#### 4.1.1 Stratégies de constitution

En stratégies de constitution, une COMET peut privilégier :

- L'adaptation par polymorphisme : cette stratégie suppose que la COMET est polymorphe. Il s'agit de jouer sur la multiplicité de ses facettes pour s'adapter au contexte d'usage. Cette stratégie conserve la COMET mais elle en ajuste l'abstraction et/ou la présentation. L'ajustement peut s'ancrer à différents niveaux d'abstraction. L'adaptation peut se faire selon différentes cardinalités :
  - 1-1 : une facette de la COMET se substitue à la facette active ;
  - 1-N : la facette active est remplacée par la complémentarité de N autres ;
  - N-1 : N facettes actives sont remplacées par une seule ;
  - N-M : N facettes actives sont remplacées par M autres.
- L'adaptation par substitution: Cette stratégie ne conserve pas la COMET. Elle la substitue par d'autres COMETs selon les cardinalités présentées ci-dessus. C'est seulement le grain d'analyse (de la facette à la COMET) qui change ;
- L'adaptation par ajout : une nouvelle COMET est introduite au système interactif afin de permettre une nouvelle tâche ou de renforcer une information (redondance de l'information) ;
- L'adaptation par suppression : une COMET est soustraite au système intractif parce que la tâche supportée par la COMET ne fait, par exemple, plus sens dans le contexte d'usage courant.

Ces stratégies de constitution s'accordent avec des stratégies de valeur.

### 4.1.2 Stratégies de valeur

[Florins 06] introduit le concept de dégradation élégante. Dans ce même esprit, mais en l'appliquant cette fois aux IHM et non plus à la plate-forme, nous identifions trois stratégies de valeur :

- L'adaptation par augmentation : cette stratégie encourage l'enrichissement fonctionnel ou non fonctionnel de l'IHM. Nous distinguons l'augmentation :
  - Fonctionnelle qui consiste à offrir des tâches utilisateur supplémentaires même si elles ne sont pas utiles. Il est intéressant de noter que cette stratégie peut en corollaire dégrader non fonctionnellement l'IHM. Elle va typiquement en conflit avec le critère de *Charge de travail-Brièveté concision* ;
  - Non fonctionnelle en termes de propriétés : si des qualités non exigées peuvent être satisfaites, alors la COMET privilégiera ces options ;
  - Non fonctionnelle dans la couverture du contexte d'usage : la COMET choisira la combinaison de facettes lui assurant un domaine de plasticité le plus large possible pour les propriétés requises.
- L'adaptation par dégradation est le dual. Dans ce cas l'IHM se dégrade fonctionnellement en supprimant des tâches déjà offertes à l'utilisateur. Une dégradation non fonctionnelle aboutira à une dégradation de la qualité en termes de propriétés
- L'adaptation par équivalence vise, au contraire, la stabilité.

À l'exécution, les stratégies sont déployées selon des politiques. Les politiques font l'objet de la section suivante.

## 4.2 Politiques

Les politiques que nous définissons sont relatives à l'autonomie de la COMET dans son adaptation. L'adaptation peut être concertée ou non concertée. La concertation peut se faire avec l'utilisateur et/ou le système environnant (COMETs ou non). Dans une version optimiste de l'adaptation concertée, l'adaptation sera appliquée sans qu'un consensus entre acteurs ne soit encore établi. La COMET peut exécuter l'adaptation sans attendre l'accord du tiers. Dans une version pessimiste, l'entendement entre acteurs sera nécessaire pour que l'adaptation soit mise en œuvre.

En cas de non concertation, l'adaptation peut être interne à la COMET auquel cas elle se chargera de la détection du contexte, calcul et mise en œuvre de l'adaptation ou, au contraire, dictée par un tiers : le système ou l'utilisateur. La Figure 97 résume cet espace problème.

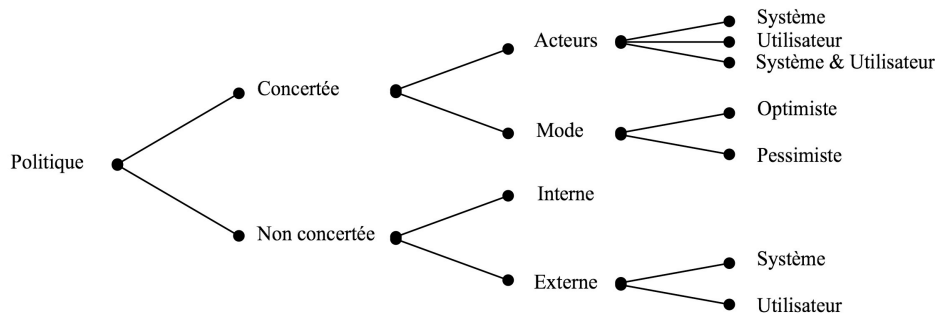


Figure 97 : Espace de classification des politiques d'adaptation.

Bien entendu, le choix de la politique dépendra, en particulier, de la capacité de la COMET à assumer ou non son adaptation.

La section suivante traite de l'architecture logicielle.



## 5. Architecture logicielle : Compact

Le modèle Compact (COntext Mouldable PAC for Plasticity) est une instantiation du modèle PAC [Coutaz 87] pour la plasticité. PAC est un modèle à agents reconnaissant trois facettes aux constituants des systèmes interactifs : une abstraction, une présentation et un contrôle. Le contrôle assure la cohérence et la communication entre les facettes abstraction et présentation. Au regard du principe de « séparation des préoccupations » assésé par le génie logiciel, Compact tranche chaque facette en deux, isolant ainsi une partie logique et une partie physique comme le montre la Figure 98.

La partie logique gère l'état de la facette indépendamment de son incarnation physique. Elle connaît, en revanche, l'ensemble des incarnations possibles (plusieurs en cas de polymorphisme) et embarque les mécanismes nécessaires à leur exploitation (commutation entre facettes, complémentarité, redondance).

La partie physique est l'incarnation effective de la COMET.

Appliquée à la facette abstraction, cette scission donne lieu à une abstraction logique et une abstraction physique. A l'instar de l'adaptateur de noyau fonctionnel dans Arch, l'abstraction logique s'apparente à une API de l'abstraction physique. L'abstraction logique embarque les mécanismes nécessaires à la commutation entre abstractions. Elle maintient l'état courant de la COMET.

De façon symétrique, la présentation logique est une API de la présentation physique. Elle gère l'ensemble des présentations physiques aux différents niveaux d'abstraction et offre les mécanismes nécessaires à leur exploitation.

Dans le même esprit, le contrôle logique gère la cohérence et la communication entre les abstraction et présentation logiques. Il représente la logique de plasticité. En pratique il est mis en œuvre par un contrôle physique qui représente le moteur de la plasticité. Le contrôle physique est en charge de :

- Recevoir et/ou capter et/ou diffuser le contexte d'usage ;
- Recevoir et/ou calculer et/ou diffuser la réaction à mettre en œuvre ;
- Eventuellement mettre en œuvre la réaction.

La réaction peut consister à changer d'abstraction ou de présentation physique. Le calcul s'appuie sur la description de la COMET. La facette Adaptation fixe les stratégies et politiques à appliquer.

A un instant donné, plusieurs abstractions et/ou présentations physiques peuvent être instanciées alors que l'instanciation est unique au niveau logique.

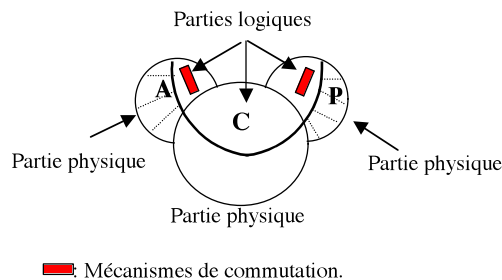


Figure 98 : Le modèle d'architecture logicielle Compact.

Comme dans PAC, un système interactif devient une collection d'agents Compact. Des canaux de communication spécifiques peuvent être établis entre les contrôles physiques pour une propagation *particulière* du contexte d'usage : par exemple, une gestion centralisée. Les contrôles plastiques sont alors tous reliés à un père calculant et propageant le nouveau contexte d'usage. Le choix de la réaction peut, de la même façon, être géré de façon centralisée.

En synthèse, nous rappelons les lignes de force des COMETs et montrons comment le concept répond aux exigences de la plasticité.

—

## 6. Synthèse

Sur un autre propos, [Beaudouin-Lafon 00] mettait en exergue le polymorphisme, la réification et la réutilisation. Ce sont ces trois mêmes arguments que nous retrouvons finalement à la base des COMETs avec une exigence supplémentaire et ici fondamentale : l'autodescription.

La motivation des COMETs était la *réutilisation* d'interacteurs façonnés pour l'adaptation. Façonnés signifie, au moins, autodécrits, peut-être polymorphes et auto-adaptatifs. Le *polymorphisme* peut s'opérer à différents niveaux d'abstraction. La *réification* est la concrétisation progressive de la COMET. Cette concrétisation est le fruit de transformations dirigées par la valeur que la COMET se doit de préserver dans le contexte d'usage courant.

La Figure 99 montre le potentiel d'une COMET à couvrir les exigences énoncées au Chapitre I. La complétude n'est pas un requis : des heuristiques sont maintenant nécessaires pour comprendre la juste balance et, en conséquence, mieux façonner les interacteurs.

Dans le modèle de la Figure 96 nous voyons la couverture des COMETS à la grille d'analyse. L'autodescription :

- Fonctionnelle : couverte par les facettes de la COMET ;
- Ontologique : correspond à start(), stop() (méthodes dans la classe COMET et structure et API dans la classe Interface finale ;
- Génétique : représentée par la getHistorique() de la classe COMET et la facette Adaptation ;
- Valeur : modélisée par la classe Valeur, propriétés, contexte d'usage et coût.

Dans le potentiel d'adaptation, le polymorphisme, s'il existe, peut être préfabriqué ou généré. Les mécanismes d'adaptation peuvent être internes et/ou externes à la COMET. La COMET décrit son contrôle d'exécution.

Approches	Autodescription									Potentiel d'adaptation						
	Fonctionnelle		Ontologique			Génétique		Valeur		Polymorphisme		Mécanismes d'adaptation			Contrôle d'exécution	
	Fonctions métier	Présentations	Etat	Structure	API	Historique	Adaptation	Ressources	Propriétés	Préfabriqué	Généré	Détection delta ctxt	Calcul de réaction	Mise en œuvre	Gestion d'état	Gestion d'exécution
COMET	x	x	x	x	x	x	x	x	x	x	x	I, E	I, E	I, E	x	x

Figure 99 : Caractérisation du concept de COMET selon la grille d'analyse.

Nous proposons dans le chapitre suivant un démonstrateur DECOR mettant en action les COMETs.

## CHAPITRE V

### LES COMETS EN ACTION : DEMONSTRATEUR DECOR

---

DECOR (DEmonstrateur de COMets en Remodelage) est une galerie numérique de peintures, multimodale et plastique, implémentée en COMETS Swing. La première section décrit le cas d'étude du point de vue de l'usage. Les sections suivantes adoptent une perspective système. Les comets sont, dans un premier temps, énumérées puis l'attention est portée sur l'architecture logicielle. En synthèse, nous portons un œil critique sur ce premier prototype.

---

#### 1. Cas d'étude

DECOR permet à un usager de parcourir un ensemble d'œuvres et d'obtenir des informations détaillées sur chacune d'elles. Les informations sont de deux natures : renseignements complémentaires sur l'œuvre et zoom sur une zone d'intérêt de l'image. Le démonstrateur tourne sur PC mais se remodèle en fonction d'un contexte d'usage simulé : la présentation s'adapte en fonction des ressources d'interaction disponibles en entrée et en sortie. A défaut d'une capture système du contexte d'usage, l'utilisateur spécifie manuellement les ressources disponibles :

- En entrée : présence ou non de souris, clavier, caméra, microphone, stylet et doigt pour une interaction directe (Figure 100a) ;
- En sortie : inclinaison verticale versus horizontale de la surface d'affichage, présence ou non de haut-parleurs. La taille de la surface d'affichage est calculée par le système interactif.

L'environnement est également simulé. Il est ici réduit à l'intensité lumineuse, au volume sonore et à l'éventuelle présence d'autres individus dans le voisinage (Figure 100b). L'utilisateur n'est pas considéré.



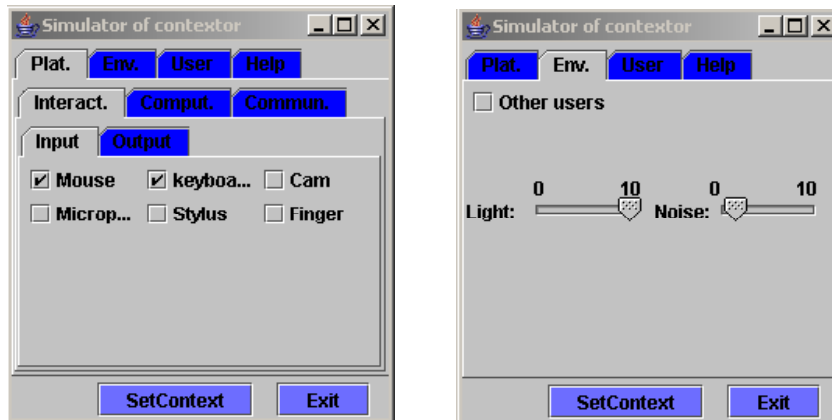


Figure 100 : Simulateur du contexte d'usage, à défaut d'une capture système.

Sur une surface verticale de grande taille, la collection des œuvres s'affiche à gauche sous la forme d'une mosaïque. Lorsque l'utilisateur sélectionne une œuvre (« La liberté guidant le peuple » sur la Figure 101), des informations détaillées sur cette œuvre s'affichent en partie droite : en haut, l'image ; en bas, des informations textuelles.

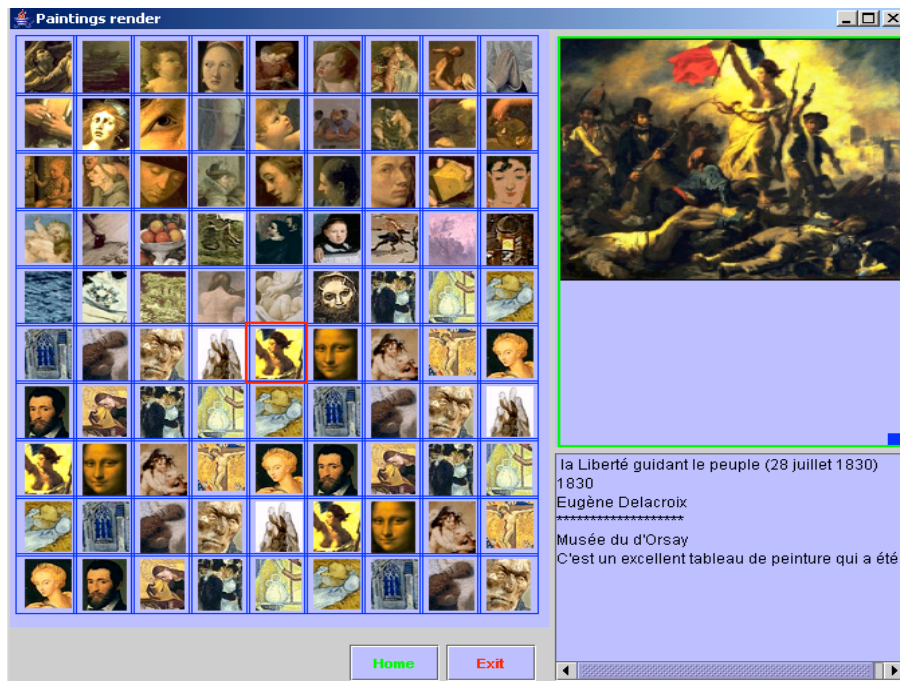


Figure 101 : DECOR en version graphique verticale.

L'utilisateur parcourt ainsi la galerie et peut, s'il le souhaite, zoomer sur une partie de l'image. Le zoom s'effectue en partie haute droite de la fenêtre (Figure 102).



Figure 102 : Zoom sur une partie de l'image, en partie haute droite de la fenêtre.

Si l'utilisateur, alors qu'il examinait le tableau de Bartolome Esteban Murillo « Le Jeune Mendiant » rétrécit la fenêtre en largeur, alors la mosaïque se transforme en liste, mais l'état d'interaction est conservé : le tableau reste sélectionné (Figure 103).



Figure 103 : La mosaïque se remodèle en liste tout en conservant l'état d'interaction.

Si l'utilisateur rétrécit cette fois la fenêtre en hauteur, alors la zone de texte affichant des informations complémentaires sur le tableau se transforme en deux boutons permettant d'entendre (et non plus de lire) les commentaires du guide (Figure 104).

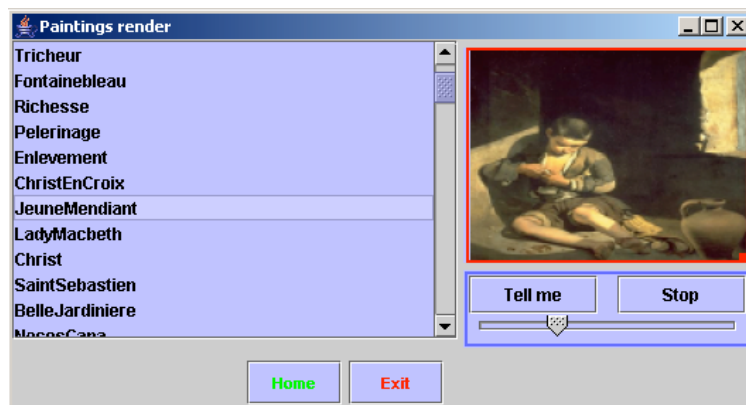


Figure 104 : Les commentaires sur le tableau sont émis en vocal dès lors que la taille de la fenêtre devient insuffisante à leur affichage graphique.

Si l'utilisateur poursuit son rétrécissement de la fenêtre, alors la liste se transforme en menu déroulant et rejoint le bandeau haut de la fenêtre (Figure 105).



Figure 105 : Au-delà d'un certain rétrécissement, la liste se compacte en un menu déroulant qui rejoint le traditionnel bandeau de la fenêtre.

Si maintenant l'utilisateur affiche la galerie sur une surface horizontale, alors une présentation circulaire est préférée permettant à tout individu situé autour de la table de profiter de la galerie. Tel un plateau tournant, les tableaux circulent devant les utilisateurs (Figure 106). Les notions de coins haut, bas, droit et gauche ne font donc plus sens. Aussi, les informations détaillées sont-elles placées au centre de la table.



Figure 106 : Présentation circulaire pour une surface d'affichage horizontale.

Si la surface d'affichage était rétrécie, alors les images seraient remplacées par du texte (Figure 107).

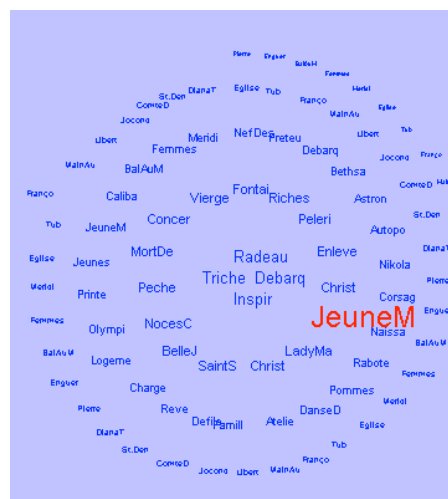


Figure 107 : Remplacement des images par du texte selon la taille de la surface d'affichage.

Enfin, terminons par la méta-IHM qui permet à l'utilisateur d'inspecter tout élément de l'IHM :

- En version non tissée, c'est une fenêtre dédiée qui rend observables les facettes de l'IHM à tout niveau d'abstraction (Figure 107). A plus haut niveau (le niveau Abstraction), l'utilisateur a accès au modèle des tâches-concepts, au modèle du contexte d'usage et aux propriétés d'ergonomie vérifiées. Cette fenêtre donne, par ailleurs, accès (en partie basse de la fenêtre) à une lentille magique. Cette lentille est à placer sur la galerie de peintures. Elle donne accès à des informations cachées. Elle peut être opaque ou transparente, rectangulaire ou ronde. L'utilisateur peut afficher ou masquer la lentille. A noter qu'il est aussi possible de migrer sur PDA la simulation du contexte. Mais, dans ce démonstrateur, l'effort est vraiment porté sur le remodelage, en particulier, multimodal ;

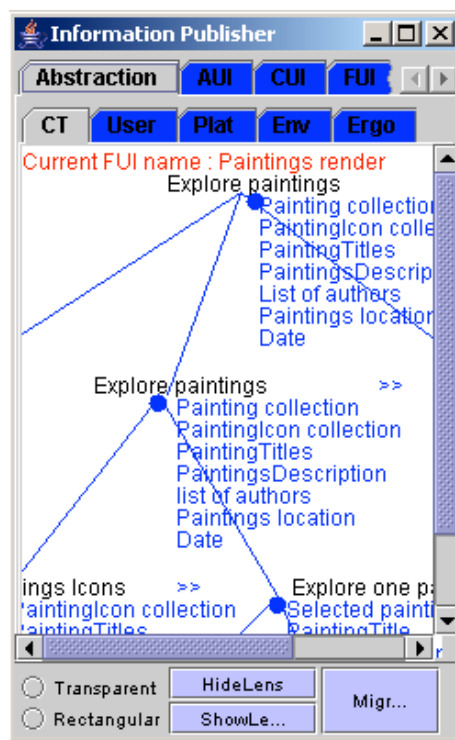


Figure 108 : Méta-IHM non tissée.

- En version tissée, la méta-IHM est la lentille magique (Figure 108). Elle affiche ici simplement le nom de la COMET en superposition de l'IHM, mais on pourrait imaginer une aide contextuelle basée sur le modèle des tâches qu'embarque l'IHM.



Figure 109 : Méta-IHM tissée.

Le cas d'étude étant présenté, la section suivante analyse le système interactif en termes de COMETS.

—

## 2. COMETS

Nous répertorions ici les COMETS de l'ordre de l'IHM puis de la méta-IHM.

### 2.1 COMETS IHM

Toutes les COMETS sont ici conformes au modèle d'architecture logicielle COMPACT. Elles gèrent leur état d'interaction, assurant ainsi une continuité de l'interaction.

## ☛ Libellé

La COMET Libellé est polymorphe dès le niveau Interface concrète. Elle peut s'afficher sous la forme d'un texte ou d'un texte condensé.

Dans DECOR, le nom de l'œuvre est un libellé. Il s'affiche, par exemple, de façon complète dans la liste (Figure 110a) alors qu'il est condensé en version circulaire (Figure 110b).

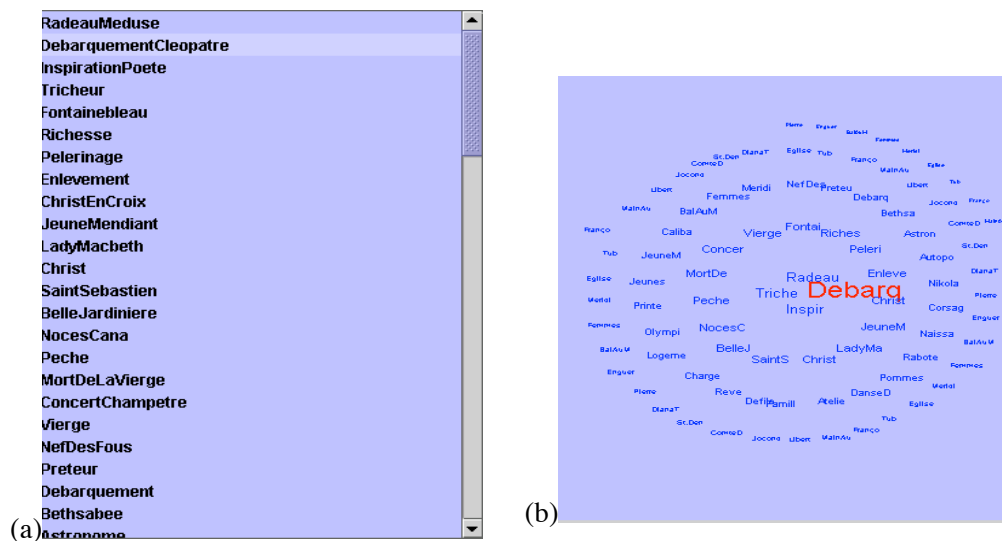


Figure 110 : COMET Libellé : version longue en a), abrégée en b).

## ☛ Texte

La COMET Texte est polymorphe dès le niveau Interface concrète. Il peut s'afficher sous la forme d'un texte ou d'un message vocal contrôlable par deux boutons « Tell me » et « Stop » ainsi qu'un potentiomètre pour le réglage du niveau sonore.

Le commentaire donné sur un tableau est un texte (Figure 111 a et b).

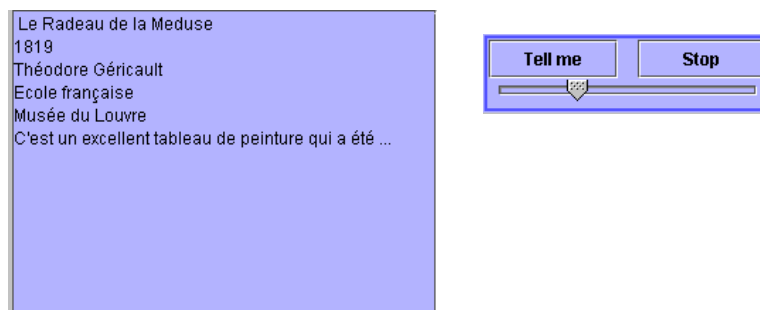


Figure 111 : COMET Texte : version monodale en a), multimodale en b).

### ☛ Image inspectable

La COMET Image inspectable est polymorphe dès le niveau Tâches-Concepts. Elle permet à l'utilisateur de consulter l'image ou l'inspecter selon la place disponible. Cette COMET est inspirée de la lentille magique en visualisation de données [Stone 94]. Elle permet la sélection d'une portion de l'image du tableau pour l'afficher dans un espace distinct. Cette tâche suppose une certaine surface d'affichage et n'est donc pas toujours possible.

La tâche est placée sous le contrôle de l'utilisateur qui peut l'inhiber ou, au contraire, l'activer. Le contrôle se fait via le carré vert situé au bas droit de l'image. Un carré vert signifie que la sélection est autorisée (Figure 112a). Dans le cas contraire, le carré est rouge (Figure 112b).





*Figure 112 : COMET Image inspectable : l'inspection n'est pas toujours possible. Sa disponibilité est rendue observable via le carré vert (a)/rouge (b).*

### ☛ Tableau de peinture

La COMET tableau de peinture est une COMET métier. Un tableau de peinture est modélisé par :

- un Libellé : le nom de l'œuvre ;
- une Image : la photo de l'œuvre ;
- un Texte : des informations détaillées.

La COMET tableau de peinture est une COMET composite gérant les COMETs Libellé, Image inspectable et Texte correspondantes. Elle s'appuie sur le polymorphisme de ses filles.

Dans DECOR, elle s'affiche sous la forme :

- Du libellé au sein de la collection de tableaux (Figure 110) ;
- De l'image pour l'inspection visuelle du tableau (Figure 112) ;
- Du texte pour les informations détaillées (Figure 111).

### ☛ Collection d'objets

La COMET collection d'objets est polymorphe dès le niveau Interface concrète. Elle dispose de cinq présentations :

- Une forme matricielle (Figure 113a) ;
- Une forme circulaire (Figure 113b) ;

- Un affichage en liste scrollable (Figure 113c) ;
- Un affichage en liste déroulante (Figure 113d).

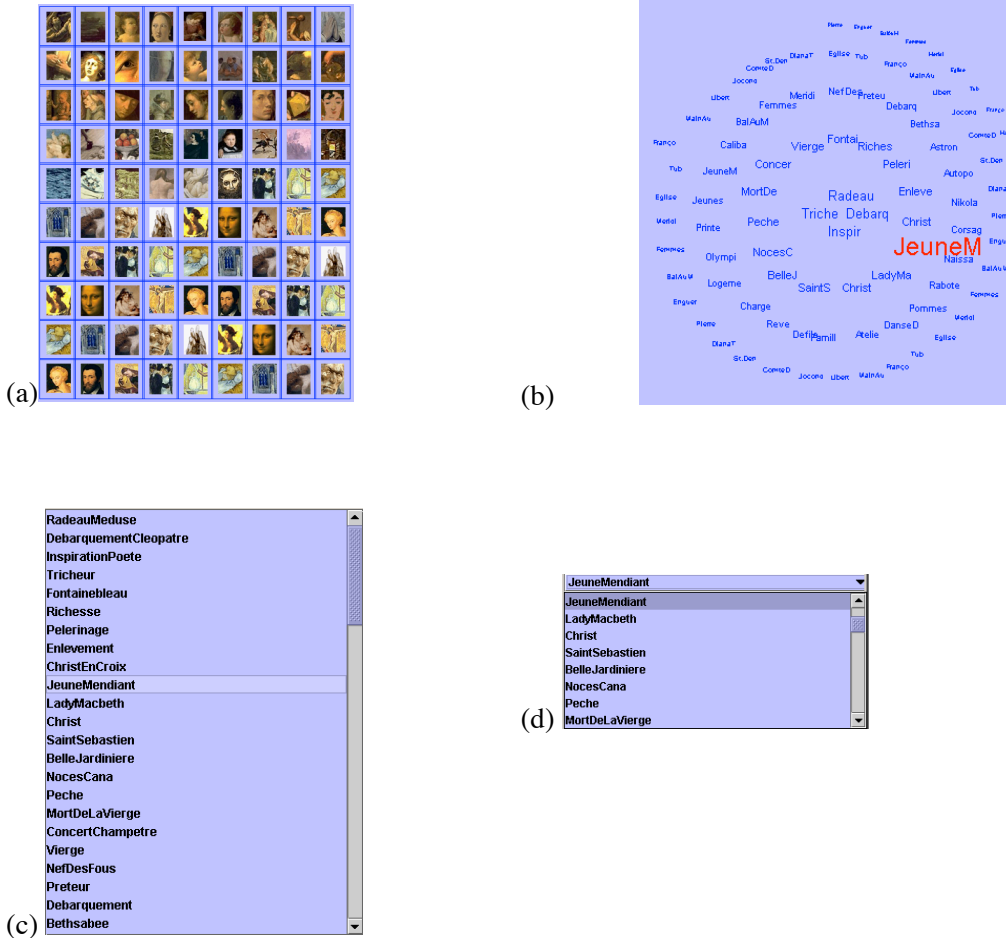


Figure 113 : COMET Collection d'objets : quatre interfaces concrètes.

### 👉 Inspecteur d'objets

La COMET Inspecteur d'objets permet d'inspecter séquentiellement des objets dans une collection. Elle comporte deux interfaces concrètes, structurées toutes deux en trois espaces de travail : un pour l'observation de la collection et la sélection d'un objet ; un autre pour l'inspection de l'objet sélectionné ; un troisième pour l'accès à des commandes.

En surface verticale, l'espace de travail associé à la collection est à gauche, tandis que l'inspection se fait à droite. Les commandes sont en bas de la fenêtre (Figure 114a). En surface horizontale, la collection est périphérique. Viennent ensuite les commandes puis l'inspection qui se fait en son centre (Figure 114b).

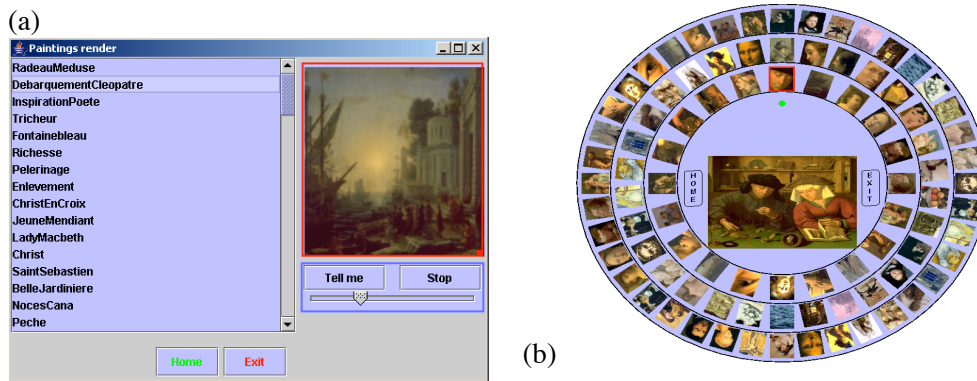


Figure 114 : COMET Inspecteur d'objets : deux présentations variant dans l'agencement des espaces de travail.

### ☛ Bouton

La COMET Bouton est polymorphe au niveau Interface concrète. Elle se présente sous une forme rectangulaire ou ovale. En version ovale, le texte est remplacé par une pastille dont la couleur est celle du libellé (Figure 115). En conséquence, la version ovale pêche sur critère de Signifiante des codes et dénominations tel que défini dans le référentiel de C. Bastien et D. Scapin. C'est à l'utilisateur de se souvenir de la sémantique de chaque couleur (vert = Home ; rouge = Exit), nuisant en conséquence à sa Charge de Travail et la Protection contre les erreurs.



Figure 115 : COMET Bouton : deux présentations concrètes.

## 2.2 COMETS Méta-IHM

### ☛ Lentille magique

La COMET Lentille magique (Figure 116) inspecte l'objet au-dessus duquel elle est située et affiche un descriptif de cet objet. Aujourd'hui, le polymorphisme se limite à de la représentation multiple : lentille rectangulaire versus ronde, opaque versus transparente, le choix étant placé sous le contrôle de l'utilisateur (Figure 108).

A terme, l'information affichée pourrait s'adapter à l'utilisateur et aux caractéristiques des objets situés en dessous :

- A l'utilisateur, pour adapter l'information et sa représentation aux compétences de ce dernier. Typiquement, si le modèle des tâches fait sens à un spécialiste IHM, il est peu probable qu'une telle représentation soit adaptée à un usager non formé à l'IHM ;
- Aux caractéristiques des objets, pour assurer la lisibilité (critère de Guidage chez C. Bastien et D. Scapin) de l'information affichée.



Figure 116 : Comet Lentille magique ici circulaire et transparente.

### ☛ Inspecteur non tissé

Aujourd'hui, dans DECOR, l'inspecteur non tissé n'est pas implémenté en COMETs. Ce que nous envisageons, c'est l'implémentation en COMETs de chaque facette et liens entre facettes. Ainsi, la lentille magique sera-t-elle tout simplement le lieu d'affichage de l'une - ou l'autre - ou les - facette(s) des objets situés en dessous. Les COMETs méta-IHM ne seraient alors plus les inspecteurs mais les inspectés.

La section suivante traite de l'architecture logicielle.



## 3. Architecture logicielle

DECOR est une COMET gérant deux filles : un inspecteur d'objets et un inspecteur non tissé. L'inspecteur d'objets gère la collection d'objets, le tableau de peinture sélectionné, les deux boutons permettant de piloter le système interactif (Home et

Exit) ainsi que la lentille magique. La collection gère l'ensemble des tableaux. Un tableau gère un libellé, un texte et une image inspectable.

C'est la COMET DECOR qui reçoit le contexte d'usage via le simulateur de contexte d'usage. Elle le propage à ses filles qui récursivement le propagent aux leurs.

Chaque COMET fonctionne par stratégie de polymorphisme, selon une politique Non concertée interne. Les autres politiques ne sont pas implémentées.



## **4. Synthèse**

En synthèse, nous dirons que ce premier prototype était plus l'occasion d'analyser un système interactif et d'identifier les « bonnes » COMETs que de valider l'approche et en extraire des heuristiques. Le contexte d'usage est simulé grâce à la comet simulateur de contexte; les descriptions sont rudimentaires limités à quelques dispositifs d'interaction; les COMETs sont homogènes dans leurs stratégies et politiques. La route est encore longue : la thèse ouvre de nombreuses perspectives.

## CONCLUSION ET PERSPECTIVES

---

Le credo défendu dans cette thèse est l'insuffisance des outils actuels pour l'ingénierie d'IHM plastiques. Si, avec du temps, un concepteur peut façonner des IHM sur mesure et gérer les changements de contexte d'usage, l'approche tombe dès lors que les contextes d'usage ne peuvent plus être prévus à la conception. Avec l'informatique ambiante, les contextes d'usage se promettent variés, variables et imprévisibles. Il faut donc envisager de nouvelles approches, de nouveaux outils. Cette thèse se situe dans le domaine de l'ingénierie de l'Interaction Homme-Machine. Elle propose une nouvelle génération d'interacteurs pour la plasticité des IHM. Nous en rappelons les contributions dans une première section puis ouvrons sur de nombreuses perspectives.

---

### 1. Contributions

A partir d'une vision des interacteurs pour la plasticité, nous avons formulé un ensemble d'exigences qui ont nourri une grille d'analyse. Cette grille d'analyse se structure en deux parties : l'une relative à la capacité d'auto-description de l'interacteur ; l'autre à son potentiel d'adaptation. Cette articulation est symptomatique du fait que la plasticité passe par la « mise au propre » de l'ingénierie des IHM traditionnelles. Aujourd'hui, si les erreurs ergonomiques sont si fréquentes, c'est peut-être parce que l'outil de construction des IHM n'incite pas le concepteur à se poser les bonnes questions. Aussi, une grande partie du travail s'est-elle attelée à mieux comprendre l'ingénierie des IHM pour mieux plastifier.

Décrire les IHM c'est comprendre et consigner leur raison d'être. Cette raison d'être est, en pratique, un ensemble de compromis conciliant des exigences d'utilisabilité avec la réalité des ressources. La notion de valeur s'est ainsi imposée. Et c'est finalement non plus l'utilisabilité que la plasticité se doit de garantir mais la valeur qu'attend l'utilisateur de son système interactif. Aussi, avons-nous amendé la définition de la plasticité et le cadre de référence CAMELEON pour intégrer cette notion de valeur.

Jusqu'ici le cadre de référence CAMELEON était méthodologique, identifiant un ensemble d'étapes dans le processus de construction d'IHM. Désormais, conception

et exécution sont décloisonnées : les niveaux d'abstraction nourrissent des facettes vivantes à l'exécution. Ces facettes racontent leur raison d'être en termes des propriétés et ressources ayant présidé à leur genèse. Elles sont la clé de l'auto-description. Elles peuvent être multiples conférant alors à l'interacteur la propriété de polymorphisme. L'interacteur peut être doué d'adaptation. Mais pour être COMET, le seul requis est l'auto-description.

Appliquer le cadre de référence CAMELEON au grain des interacteurs, c'est implicitement reconnaître que le grain d'une COMET n'est pas fixé : elle peut, en conséquence, aller du champ texte à un système interactif complexe. Cette ouverture dissuade d'une classification fonctionnelle fermée. Il faut, au contraire, ouvrir résolument sur de nouvelles tâches, au prix, peut-être, d'ontologies et d'alignements entre ontologies. En particulier, dès lors que la plasticité est placée sous le contrôle de l'utilisateur, de nouvelles tâches apparaissent : inspecter, remodeler, migrer, etc. Ces tâches n'avaient jusqu'ici encore jamais été envisagées. Elles relèvent de la méta-IHM.

La méta-IHM tend à briser les frontières entre concepteur et utilisateur. Si cette frontière saute, alors la plasticité appellera la plasticité pour adapter la méta-IHM aux compétences de l'utilisateur : c'est la plasticité de la méta-IHM comme évoqué dans le démonstrateur DECOR.

DECOR est un premier prototype. Le modèle conceptuel des COMETs n'y est que partiellement illustré. Les perspectives sont nombreuses tant au plan conceptuel qu'implémentational. L'implémentation sera nécessaire pour encore mieux comprendre et, en conséquence, pouvoir établir des heuristiques notamment sur les stratégies et politiques à promouvoir.



## 2. Perspectives

L'ambition était grande : modéliser et évaluer la valeur, c'est palper le rapport entre la satisfaction du besoin et le coût. Le palper d'un point de vue système ! Il nous semble aujourd'hui que la voie est tracée. La thèse a permis de bien comprendre le problème, d'en poser les fondements (classifications et modèles). Il s'agit maintenant d'aller en profondeur et décrire chaque facette.

L'Ingénierie Dirigée par les Modèles (IDM) apporte un savoir-faire en description et des outils généraux pour leur transformation. Le rapprochement avec cette communauté est amorcé [Sottet 06]. Des méta-modèles sont en cours d'élaboration mais, là encore, la voie est tout juste tracée. En graphique, ils ne sont toujours pas stabilisés. Les autres modalités et la multimodalité sont des perspectives copieuses. Un langage à suivre et peut-être influencer est UsiXML (<http://www.usixml.org>).

Au-delà de la description, se posent des problèmes d'adaptation. Il nous faut des compétences système pour percevoir le contexte d'usage. Il nous faut des heuristiques pour régir l'adaptation des COMETS. Des stratégies et politiques sont certes identifiées. Mais en pratique, quelle est la valeur de ces stratégies ? En particulier, quelle est la « bonne » concertation au sein de la hiérarchie de COMETS ?

Si raisonner en monde fermé est déjà complexe, qu'en est-il en monde ouvert où les COMETS se découvrent, se composent ? En IDM, le zoo de méta-modèles (projet zoomm pour Zoo Of MetaModels) apportera une pierre à l'édifice mais la composition d'IHM pose des problèmes spécifiques. Par exemple, si des concepts sont partagés par deux COMETS, ils devront être factorisés au risque sinon de dégrader la Charge de travail dans son sous-critère Brièveté tel qu'exprimé dans le référentiel de C. Bastien et D. Scapin. La factorisation peut se répercuter par la création d'un nouvel espace de travail, imposant en conséquence une révision complète des IHMs. Là encore, on perçoit bien que la plasticité ne sera réalité que quand le savoir-faire traditionnel sera consigné et formalisé.

A terme, nous visons un atelier de plasturgie. La route est encore longue en conception, implémentation et évaluation, surtout si l'utilisateur devient créateur de son IHM et donc peut-être candidat à l'utilisation de cet atelier à l'exécution.





## BIBLIOGRAPHIE

---

- [Antoniac 02] : Antoniac, P., Pulli, P., Kuroda, T., Bendas, D., Hickey, S., Sasaki, H. Wireless User Perspectives in Europe: HandSmart Mediaphone Interface, Wireless Personal Communications, Vol. 22, pp. 161-174, 2002.
- [Balbo 94] : Balbo, S. Evaluation Ergonomique des Interfaces Utilisateur : Un Pas Vers l'Automatisation, Thèse de l'Université Joseph Fourier de Grenoble, Septembre 1994, 287 pages.
- [Ballagas 03] : Ballagas, R., Ringel, M., Stone, M., Borchers, J. Between u and i: iStuff: a physical user interface toolkit for ubiquitous computing environments, Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'03), 2003, April 5-10, Ft. Lauderdale, Florida, USA, pp 537-544.
- [Barnard 85] : Barnard, P.J. Interacting Cognitive Subsystems: A psycholinguistic approach to short term memory. In: Ellis A (ed) Progress in the psychology of language. Lawrence Erlbaum, London 1985, pp 197-258.
- [Barouillet 96] : Barouillet, P. Ressources, capacités cognitives et mémoire de travail : postulats, métaphores et modèles. Psychologie Française. n° spécial : A. Tricot & L. Chanquoy (Eds). La charge mentale, 1996.
- [Bastien 93] : Bastien, J.M.C., Scapin, D.L. Critères ergonomiques pour l'évaluation d'interfaces utilisateurs. Rapport technique INRIA n°156, Juin 1993, INRIA Le Chesnay.
- [Beaudouin-Lafon 00] : Michel Beaudouin-Lafon et Wendy Mackay. Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces. In Proc. Advanced Visual Interfaces, AVI 2000, Palerme (Italie), Mai 2000, ACM Press, pp 102-109.
- [Bérard 99] : Bérard, F. The Perceptual Window: Head Motion as a new Input Stream. In Proc. Interact99, Edinburgh, A. Sasse & C. Johnson Eds, IFIP IOS Press Publ., 1999, pp. 238-244.
- [Beugnard 99] : Beugnard, A., Jézéquel, J.M., Plouzeau, N., Watkins, D. Making Components Contract Aware, Computer, July 1999, pp 38-45.
- [Bier 93] : Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D. Toolglass and Magic Lenses: The See-Through Interface, ACM 1993.
- [Blattner 89] : Blattner, M., Sumikawa, D., Greenberg, R. Earcons, icons : their structure, common design principles. In Proc. HCI'89, 1989.
- [Blattner 96] : Blattner, M.M., Glinert, E.P. Multimodal Integration, IEEE MultiMedia, Volume 3, Issue 4, December 1996, pp 14-24.
- [Boy 89] : Boy, G. Intelligent Assistant Systems. T. J. Press (Padstow) Limited, Padstow, Cornwall, 1989, pp 250-251.

- [Brewster 97] : Brewster, S.A., Clarke, C.V. The design and evaluation of a sonically-enhanced tool palette, Proceedings of ICAD'97 (Xerox PARC, USA), Xerox, 1997, pp 119-123.
- [Brisson 94] : Brisson, G., Andre, J. PROSPECT. Analyse et spécification de l'interface utilisateur d'un système interactif, Actes du colloque ERGO.IA'94, Biarritz, 1994, pp 382-393.
- [Broekhuizen 03] : Broekhuizen, T.L.J., Jager, W. A conceptual model of channel choice: Measuring online and offline shopping value perceptions, University of Groningen, Research Institute SOM (Systems, Organisations and Management), Research Report 04F04, 2003. Disponible à <http://www.bdk.rug.nl/medewerkers/t.l.j.broekhuizen>
- [Buxton 83] : Buxton, W. Lexical and pragmatic considerations of input structures, Computer Graphics 17 (1), pp 31-37, 1983.
- [Calvary 01] : Calvary, G., Coutaz, J., Thevenin, D. A Unifying Reference Framework for the Development of Plastic User Interfaces, Proceedings of 8<sup>th</sup> IFIP International Conference on Engineering for Human-Computer Interaction EHCI'2001 (Toronto, 11-13 May 2001), R. Little and L. Nigay (eds.), Lecture Notes in Computer Science, Vol. 2254, Springer-Verlag, Berlin, 2001, pp. 173-192.
- [Calvary 03] : Calvary, G., Coutaz J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A unifying reference framework for multi-target user interfaces, Interacting With Computers, Vol. 15/3, pp 289-308, 2003.
- [Calvary 04] : Calvary, G., Coutaz, J., Dâassi, O., Balme, L., Demeure, A. Towards a new generation of widgets for supporting software plasticity: the « comet », EHCI-DSVIS'2004, The 9th IFIP Working Conference on Engineering for Human-Computer Interaction Jointly with The 11th International Workshop on Design, Specification and Verification of Interactive Systems, Bastide, R., Palanque, P., Roth, J. (Eds), Lecture Notes in Computer Science 3425, Springer, ISSN 0302-9743, Hamburg, Germany, July 11-13, 2004, pp 306-323.
- [Calvary 05] : Calvary, G., Daassi, O., Coutaz, J., Demeure, A. Des Widgets aux comets pour la Plasticité des Systèmes Interactifs. Revue d'Interaction Homme-Machine, Europa, Paris, Volume 6, n°1, ISSN 1289-2963, pp 33-53, 2005.
- [Calvary 06] : Calvary, G., Coutaz, J., Dâassi, O., Ganneau, V., Balme, L., Demeure, A., Sottet, J-S. Métamorphose des IHM et Plasticité : Article de synthèse, 10ème conférence ERGO-IA, L'humain comme facteur de performance des systèmes complexes, 11-13 Octobre 2006, E. Brangier, C. Kolski et J.R. Ruault (eds), ESTIA & ESTIA.INNOVATION, Biarritz, France, pp 79-86.
- [Calvillo-Gamez 03] : Calvillo-Gamez, E.H., Leland, N., Shaer, O., Jacob, R.J.K. The TAC Paradigm: Unified Conceptual Framework to Represent Tangible User Interfaces, ACM International Conference Proceeding Series; Vol. 46, Proceedings of the Latin American conference on Human-computer interaction, 2003, Rio de Janeiro, Brazil, pp 9-15.

- [Card 83] : Card, S., Moran, T., Newell, A. The psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, 1983.
- [Chevallier 89] : Chevallier, J. Produits et Analyse de la valeur, AV et analyse fonctionnelle : deux clés pour un produit performant, Collection Automatisation et Production, Cepadues-Editions, ISBN 2.85428.189.6, 184 pages, 1989.
- [Cheverst 01] : Cheverst, K., Davies, N., Mitchell, K., Efstratiou, C.: Using context as a Crystal Ball: Rewards and Pitfalls, Personal and Ubiquitous Computing, Springer Ed., 5(1), 2001, pp. 8-11.
- [Chung 91] : Chung, L. Representation and Utilization of Non-functional Requirements for Information System Design. In 3<sup>rd</sup> international conference on Advanced Information System Engineering – CaiSE'91, Trondheim, Norway, May 1991.
- [Constantine 99] Constantine, L. Software for Use, Addison-Wesley, 1999, pp. 427
- [Coutaz 87] : Coutaz, J. PAC, an Object Oriented Model for Dialog Design, In INTERACT 87 - 2nd IFIP International Conference on Human-Computer Interaction, September 1-4, 1987, Stuttgart, Germany, H.J. Bullinger, B. Shackel (Eds.), pp 431-436.
- [Crease 01] : Crease, M. A Toolkit of Resource-Sensitive Multimodal Widgets, PhD Thesis, Department of Computing Science, University of Glasgow, December 2001.
- [Dâassi 03] : Dâassi, O., Calvary, G., Coutaz, J., Demeure, A. Comet : Une nouvelle génération de « Widget » pour la Plasticité des Interfaces, Actes de la 15<sup>ème</sup> conférence francophone sur l'Interaction Homme-Machine, IHM 2003, Caen, France, Novembre 2003, pp 64-71.
- [De Palma 1999] : De Palma N., Bellisard L., Riveill M. Dynamic Reconfiguration of Agent-Based Applications . Third European Research Seminar on Advances in Distributed Systems (ERSADS'99), Madeira Island (Portugal), 1999
- [Denis 04] : Denis, C., Karsenty, L. Inter-Usability of Multi-Device Systems – A Conceptual Framework, In Multiple User Interfaces – Cross-Platform Applications and Context-aware Interfaces, Seffah, A., Javahery, H. (eds), John Wiley & Sons Publ., 2004, pp 373-385.
- [Dey 99] : Dey, A.K., Salber, D., Futakawa, M., Abowd, G.D. An Architecture To Support Context-Aware Applications, GVU Technical Report GIT-GVU-99-23, June 1999.
- [Dey 00] : Dey, A. Providing Architectural Support for Building Context-Aware Applications. Thèse pour l'obtention du titre de docteur en Informatique, Institut Technologique de Géorgie (Georgia Tech), 2000.
- [Dieterich 93] : Dieterich, H., Malinowski, U., Kühme, T., Schneider-Hufschmidt, M. State of the Art in Adaptive User Interfaces, in Adaptive User Interfaces: Principles and Practice, Schneider-Hufschmidt & Al.(ed.), 1994, pp. 13-48.
- [Dubois 01] : Dubois, E. Chirurgie Augmentée, un cas de Réalité Augmentée, Conception et réalisation centrées sur l'utilisateur, Thèse de doctorat

Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (IMAG), Université Joseph Fourier, Juillet 2001, 275 pages.

- [Duke 93a] : Duke, D., Faconti, G., Harrison, M., Paterno, F. Unifying Views of Interactors, The AMODEUS Project, ESPRIT Basic Research Action 7040, D1: First theory of interactors; analysis of the relationship between the reference models, Duke, D., Harrison, M. Coutaz, J., Nigay, L., Salber, D., Faconti, G., Paterno, F., 25th May 1993.
- [Duke 93b] : Duke, D., Harrison, M. Abstract Interaction Objects, In Hubbard RJ., Juan R (Eds.) EUROGRAPHIQUICS'93, Computer Graphics Forum, Vol. 12, No.3, pp. 26-36.
- [Duke 95] : Duke, D.J., Harrison, M.D. Event Model of Human-system interaction, Software Engineering Journal, Vol. 10, No.1, 1995, pp. 3-12.
- [Englander 01] : Englander, R. Developing Java Beans, O'Reilly publ., 2001, 316 pages.
- [Faconti 90] : Faconti, G., Paterno, F. An approach to the formal specification of the components of an interaction, In C. Vandoni and D. Duce, editors, Eurographics 90, North-Holland, 1990, pp 481-494.
- [Faconti 93] : Faconti, G. Towards the Concept of Interactor, AMODEUS project report, ref. sm/wp8, 1993.
- [Fails 02] : Fails, A.J., Olsen, D. Light Widgets: Interacting in Every-day Spaces. In the proceedings of IUI'02, January 13-16, 2002, San Francisco, California, USA, pp 214-215.
- [Florins 04] : Florins, M., Trevisan, D., Vanderdonck, J. The continuity property in mixed reality and multiplatform systems : A comparative study. CADUI 2004, pp 321-332.
- [Florins 06] : Florins, M. Graceful Degradation, A Method for Designing Multiplatform Graphical User Interfaces, PhD Université Catholique de Louvain, Belgique, Juillet 2006.
- [Foley 84] : Foley, J.D., Wallace, V. L., Chan, P. The human factors of computer graphics interaction techniques, IEEE Computer Graphics Applications, 4(11), 1984, pp 13-48.
- [Frohlich 91] : Frohlich, D.M. The Design Space of Interfaces, Multimedia Systems, Interaction and Applications, Proceedings of 1st Eurographics Workshop, Stockholm, Sweden, April 18/19, 1991, Springer Verlag, pp 53-69.
- [Gram96] : Gram, C. Cockton, G. Design Principles for Interactive Software, 1996 Chapman et Hall.
- [Greenberg 01] : Greenberg, S. Phidgets: Easy Development of Physical Interfaces through Physical Widgets, Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST'01), 2001, Orlando FI-A, pp 209-218;

- [Griffiths 01] : Griffiths, T., Barclay, P.J., Paton, N.W., McKirdy, J., Kennedy, J., Gray, P.D. Teallach: A Model-Based User Interface Development Environment for Object Databases, *Interacting With Computers*, 14(1), pp 31-68.
- [Grolaux 01] : Grolaux, D., Van Roy, P., Vanderdonckt, J. FlexClock: a Plastic Clock Written in Oz with the QtK Toolkit, In: *Proceedings de TAMODIA 2002 (First International Workshop on Task Models and Diagrams for User Interface Design)*, Bucharest, 18-19 july 2002, ISBN: 973-8360-01-3, Pribeanu, C., Vanderdonckt, J. (eds), INFOREC Publishing House Bucharest, 2002, pp 135-142.
- [Halford 93] : Halford, G.S. *Children's understanding : The development of mental models*; Hillsdale, NJ, Lawrence Erlbaum Associates, 1993.
- [Hinckley 03] : Hinckley, K., Jacob, R., Ware, C. Input/Output Devices and Interaction Technics, In *CRC Computer Science and Engineering Handbook*, CRC Press, Boca Raton, FL, 2003.
- [Hinckley 04] : Hinckley, K., Jacob, R.J.K., Ware, C. Input/output Devices and Interaction Techniques, In *The Computer Science Handbook, Second Edition*, Ed. by A.B. Tucker, Chapman and Hall/CRC Press, 2004, pp 20.1-20.32
- [Ishii 98] : Ishii, H., Wisneski, C., Brave, S., Dahley, S., Gorbet, M., Ullmer, B., Yarin, P. ambientROOM: integrating ambient media with architectural space, In the proceedings of the CHI'98 conference on Human factors in computing systems, ACM Press, Los Angeles, California, 1998, pp 173-174.
- [ISO 03] : ISO/IEC CD 25000.2. Software and Systems Engineering – Software product quality requirements and evaluation (SquaRE) – Guide to SquaRE, 2003-01-13.
- [Jabarin 03] : Jabarin, B., Graham, T.C.N. Architectures for Widget-Based Plasticity, in *Proceedings of Design, Specification and Verification of Interactive Systems (DSV-IS 2003)*, Springer LNCS, 2003, pp 124-138.
- [Jacob 96] : Jacob, R.J.K. *Human-Computer Interaction: Input Devices*, ACM Computing Surveys, March 1996, Vol. 28, no. 1, pp. 177-179.
- [Harris 91] : Harris, B. *The true Art of Screenwriting*, 1991, Disponible en ligne.
- [Hart 88] Hart, S.G., Staveland, L.E. Developpement of NASA-TLX (Task Load Index) : results of experimental and theoretical resaerch. In P.A. Hancock et N. Meshkati (Eds.), *Human Mental Workload* (pp. 139-183). Amesterdam : North holland.
- [Kasabach 98] : Kasabach, C., Pacione, C., Stivoric, J., Gemperle, F., Siewiorek, D. Digital Ink: A Familiar Idea with Technological Might, In proceedings of the ACM CHI conference on Human Factors in Computing Systems (CHI'98), April 1998, Los Angeles, California, pp 175-176.
- [Kawai 96] : Kawai, S., Aida, H., Saito, T. Designing Interface Toolkit with Dynamic Selectable Modality, in *Proceedings of Assets '96, the Second International*

- ACM Conference on Assistive Technologies, Vancouver, April 1996, pp. 72-79.
- [Kitchenham 96] : Kitchenham, B., Pfleeger, S. Software Quality: The Exclusive Target, IEEE Software, January 1996, pp 12-21.
- [Lachenal 04] : Lachenal, C. Modèle et infrastructure logicielle pour l'interaction multi-surface multi-instrument. Thèse de doctorat Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (CLIPS), Université Joseph Fourier, Décembre 2004.
- [Lachnitt 80] : Lachnitt, J. L'analyse de la valeur. Presses universitaires de France 1980.
- [Le Moigne 77] : Le Moigne, J.L. La théorie du système général, théorie de la modélisation, Systèmes-Décisions, Presses universitaires de France, 1977.
- [Lopez-Jacquero 04] : Lopez-Jaquero, V., Montero, F., Molina, J.P., Gonzalez, P. A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties, EHCI-DSVIS'2004, The 9th IFIP Working Conference on Engineering for Human-Computer Interaction Jointly with The 11th International Workshop on Design, Specification and Verification of Interactive Systems, Bastide, R., Palanque, P., Roth, J. (Eds), Lecture Notes in Computer Science 3425, Springer, ISSN 0302-9743, Hamburg, Germany, July 11-13, 2004, pp 289-291.
- [Markopoulos 97] : Markopoulos, P. A compositional model for the formal specification of user interface software, Submitted for the degree of Doctor of Philosophy, March 1997.
- [Massink 02] : Massink, M., Faconti, G. A reference Framework for Continuous interaction, Journal of Universal Access in the Information Society, 2002, pp 237-251.
- [Morin 90] : Morin, D. Working Group Discussion: Current Practice, in D.A. Duce, M.R. Gomes, F.R.A. Hopgood, J.R. Lee (éds.), User Interface Management and Design, Proceedings of the Workshop on User Interface Management Systems and Environment, Lisbonne, 4-6 Juin 1990, Eurographics Seminars Series, Springer-Verlag, Berlin, 1991, pp. 51-56.
- [Myers 90] : Myers, B.A. A New Model for handling Input, ACM Transactions on Information Systems, Vol.3, Np. 3, juillet 1990, pp. 289-320.
- [Myers 01] : Myers, B.A. Using Hand-Held Devices and PCs Together, Communications of the ACM, Volume 44, Issue 11, November 2001, pp 34-41.
- [Nielsen 93] : Nielsen, J. Usability Engineering. Academic press, San Diego, CA, 1993.
- [Nigay 94] Nigay, L. Conception et modélisation logicielles des systèmes interactifs: application aux interfaces multimodales. Thèse informatique à l'université Joseph-Fourier, 1994.

- [Nogier 05] : Nogier, J.F. Ergonomie du logiciel et design WEB. Le manuel des interfaces utilisateur, 3ème édition, mars 2005, Dunod, 272 pages.
- [Norman 86] : Norman, D. A., Draper, S. W. User Centered System Design, Lawrence Erlbaum Associates, Publishers, 1986.
- [Normand 92] : Normand, V. Le modèle SIROCO : de la spécification conceptuelle des interfaces utilisateur à leur réalisation, Thèse de l'Université Joseph Fourier-Granola I, Spécialité Informatique, Avril 1992, 258 pages.
- [Nylander 04] : Nylander, S., Bylund, M., Waern, A. The Ubiquitous Interactor–Device Independent Access to Mobile services, CADUI 2004, pp 269-280.
- [Olive 97] : Olive, T., Piolat, A., Roussey, J-Y. Effort Cognitif et mobilisation des processus en production de texte, Effet de l'habileté rédactionnelle et du niveau de connaissances, In A. Vom Hoffe & D. Mellier (Eds.), Attention et contrôle cognitif : Mécanismes, développement, habiletés et pathologies, Rouen : Presses Universitaires de Rouen, 1997, pp 71-85.
- [Olsen 00] : Olsen, D.R., Nielsen, S.T., Moyes, M., Fredrickson, P. Cross-modal Interaction using Xweb. Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST), San Diego, California, United States, 2000, pp 191-200.
- [Olsen 01] : Olsen, D. R., Hudson, S., Tam, C. M., Conaty, G., Phelps, M., Heiner, J.; Speech Interaction with Graphical User Interfaces, Interact 2001, IOS Press, 2001.
- [Oreizy 99] : Oreizy, P., Taylor, R. An Architecture-Based Approach to Self-Adaptive Software. In IEEE Intelligent Systems. May-June, 1999, pp. 54-62.
- [Oulasvirta 04] : Oulasvirta, A., Saariluoma, P. Long-term working memory and interrupting messages in human-computer interaction. Behaviour and Information Technology, January-February 2004, Vol. 23, No. 1, pp 53-64.
- [Paterno 97] : Paterno, F., Mancini, C., Meniconi, S. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models, Proceedings Interact'97, July'97, Sydney, Chapman&Hall pp 362-369.
- [Pérez-Quiñones 01] : Pérez-Quiñones, M.A. Capra, R.G., Shao, Z. The Ears Have It: A Task by Information Structure Taxonomy for Voice Access to Web Pages. In Proceedings Human-Computer Interaction Interact '03, pp. 856-859. IOS Press: Amsterdam. September 1-3, 2003, Zurich, Switzerland.
- [Reenskaug 79] : Reenskaug, T. Models-Views–Controllers, Xerox PARC technical note, December 1979.
- [Rekimoto 97] : Rekimoto, J. Pick and Drop: A Direct Manipulation Technique for Multiple Computer Environments, In the proceedings of the UIST'97 conference (User Interface Software and Technology), ACM Press, 1997, pp 31-39.
- [Rekimoto 99] : Rekimoto, J., Saitoh, M. Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments, In the proceedings of the CHI'99 conference on Human factors in computing systems, ACM Press, 1999, pp. 378-385.



- [Rosa 02] : Rosa, N.S., Cunha, P.R.F., Justo, G.R.R. ProcessNFL: A Language for Describing Non-Functional Properties. In Proceedings of the 35th Hawaii International Conference on System Sciences, IEEE Press, 2002.
- [Rosson 02] : Rosson, M.B., Carroll, J.M. Usability Engineering: Scenario-Based Development of Human-Computer Interaction, San Francisco, CA : Morgan Kaufman, 2002.
- [Roudaut 06] : Roudaut, A., Coutaz, J. **Méta-IHM ou comment contrôler l'espace interactif ambiant**, Actes des Troisièmes Journées Francophones : Mobilité et Ubiquité 2006, Ubimob 2006, ACM Press, Paris. Accessible dans la digital Library de l'ACM. 5-8 septembre 2006.
- [Rudisill 88] : Rudisill, M., Gillan, D.J. Space Station Information System Human-Computer Interface Guide. NASA Report, Use 1000, V 2.0, May 1988.
- [Salber 95] : Salber, D. De l'interaction individuelle aux systèmes multi-utilisateurs. L'exemple de la Communication Homme-Homme-Médiatisée, Thèse de doctorat Informatique préparée au Laboratoire de Génie Informatique (IMAG), Université Joseph Fourier, 8 septembre 1995, 303 pages.
- [Schmidt 99] : Schmidt, A. Implicit human-computer interaction through context. 2th Workshop on Human Computer Interaction with Mobile Devices. Edinburgh, Scotland, 31 August 1999.
- [Schneiderman 96] : Schneiderman, B. The Eyes Have It : A Task by Data Type Taxonomy for Information Visualization, CS-TR-3665 July 1996, ISR-TR-96-66.
- [Seffah 01] : Seffah, A., Kececi, N., Donyaee, M. QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models. In Proceedings APAQS Second Asia-Pacific Conference on Quality Software, December 2001, Hong-Kong, pp 10-11.
- [Shackel 91] : Shackel, B. Usability- context, framework, design and evaluation. In Shackel, B., Richardson, S. (eds). Human Factors for Informatics Usability. Cambridge University Press, Cambridge, 1991, pp 21-38.
- [Sottet 06] : Sottet, J.S., Calvary, G., Favre, J.M. Mapping Model: A First Step to Ensure Usability for sustaining User Interface Plasticity, MDDAUI'06 held in conjunction with MoDELS'06, Geneva, Italy, October 3, 2006.
- [Speier 97] : Speier, C., Valacich, J., Vessey, I. The effects of Task Interruption and Information Presentation On Individual Decision Making. International conference on information system, Atlanta, Georgia, United States, 1997, pp 21-36.
- [Spence 01] : Spence, C., Nicholls, M.E.R., Driver, J. The cost of expecting events in the wrong sensory modality, Perception & Psychophysics, 2001, 63 (2), pp 330-336.
- [Stone 94] : Stone, M.C., Fishkin, K., Bier, E.A. The Movable Filter as a User Interface Tool, Proceedings of the SIGCHI conference on Human factors in computing systems CHI '94, 1994, pp 306-312.

- [Suchman 87] : Suchman L.A. Plans and situated actions. The problem of human-machine communication. Cambridge University Press, 1987.
- [Sweller 91] : Sweller, J., Chandler, P. Evidence for cognitive load theory. *Cognition and Instruction*, 8, 351-362, 1991.
- [Szekely 96] : Szekely P. Retrospective and Challenges for Model-Based Interface Development, Proceedings of CADUI'96 (Computer-Aided Design of User Interfaces), J. Vanderdonckt (eds), Presses Universitaires de Namur, 1996, pp xxi-xliv.
- [Tan 03] : Tan, D.S., Czerwinski, M. Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays, OZCHI 2003 Conference for the Computer-Human Interaction Special Interest Group of the ergonomics Society of Australia, pp 184-191.
- [Thevenin 99] : Thevenin, D., Coutaz, J. Plasticity of User Interfaces: Framework and Research Agenda. In Proc. Interact'99, Edinburgh, Sasse, A., Johnson, C. Eds, IFIP IOS Press Publ., 1999, pp.110-117.
- [Thevenin 01] : Thevenin, D. Adaptation en Interaction Homme-Machine : Cas de la plasticité, Thèse de doctorat de l'université Joseph Fourier, 21 Décembre 2001.
- [Trevisan 03] : Trevisan, D., Vanderdonckt, J., Macq, B. Continuity as a usability property, HCI 2003 - 10th Intl Conference on Human-Computer Interaction, Heraklion, Greece, June 22-27, 2003, Vol. I, pp 1268-1272.
- [Trevisan 04] : Trevisan, D., Vanderdonckt, J., Macq, B. Conceptualizing Mixed Spaces of Interaction for Designing Continuous Interaction, *Virtual Reality Journal*, 8, pp 83-95, 2004.
- [UIMS 92] : The UIMS Workshop Tool Developers: A Metamodel for the Runtime Architecture of an Interactive System, *SIGCHI Bulletin*, 1992.
- [Vanderdonckt 93] : Vanderdonckt, J., Bodart, F. Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection, Proceedings of the Conference on Human Factors in Computing Systems InterCHI'93, Bridges Between Worlds, Amsterdam Pays-Bas, 24-29 avril 1993, S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, T. White (éds.), ACM Press, New York, 1993, pp. 424-429.
- [Vanderdonckt 97] : Vanderdonckt, J. Conception assistée de la présentation d'une interface homme-machine ergonomique pour une application de gestion hautement interactive. Thèse présentée en vue de l'obtention du grade de docteur en science (Option Informatique), 9 juillet 1997, facultés Universitaires Notre-Dame de la paix.
- [Vernier 01] : Vernier, F. La multimodalité en sortie et son application à la visualisation de grandes quantités d'information. Thèse de doctorat Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (IMAG), Université Joseph Fourier, Février 2001, 238 pages.

[Wellner 93] : Wellner, P. Interacting with Paper on the DigitalDesk, Communications of the ACM, Vol. 36, No 7, July 1993, pp. 87-96.

[W3C 06] : <http://www.w3.org>, 1 Juin 2006