

## Chapitre 9<sup>1</sup>

# Systèmes interactifs et adaptation centrée utilisateur : la plasticité des Interfaces Homme-Machine

Publié dans *Informatique et Intelligence ambiante : des capteurs  
aux applications*, Hermes Sciences Publishing Ltd

### 9.1. Introduction

Un système interactif est un système numérique en interaction avec un humain. Il comprend deux grandes classes de fonctions : (1) un ensemble de services dépendants d'un domaine d'application : c'est le noyau fonctionnel du système ; (2) un ensemble de fonctions et d'éléments matériels servant d'intermédiaires entre le noyau fonctionnel et l'humain : c'est l'Interface Homme-Machine (IHM) du

---

1. Chapitre rédigé par Joëlle COUTAZ, Gaëlle CALVARY, Alexandre DEMEURE et Lionel BALME

## 2 Titre de l'ouvrage

système. Les éléments matériels de cette IHM comme l'écran et la souris, perceptibles et actionnables par l'utilisateur, tiennent le rôle de ressources d'interaction. L'Interaction Homme-Machine, en tant que domaine d'étude, intervient dans la « bonne construction » des systèmes interactifs.

L'Interaction Homme-Machine porte sur l'élaboration de théories, de concepts, de modèles, de méthodes et de solutions techniques pour la conception et la mise en œuvre de systèmes interactifs utiles, utilisables et plaisants, autrement dit, des systèmes à valeur ajoutée pour les utilisateurs cibles. L'utilité se mesure en termes de conformité fonctionnelle du noyau fonctionnel du système avec les services attendus par l'utilisateur cible. Par exemple, un système de chauffage doit permettre de faire des économies si cela correspond à l'attente de l'utilisateur. Pour un autre utilisateur (voire pour ce même utilisateur mais en des circonstances différentes), ce sera, non plus l'économie, mais le confort avant tout. L'utilisabilité se mesure en termes de conformité de l'IHM avec les capacités cognitives et sensori-motrices de l'utilisateur. Dans notre exemple, le système de chauffage du premier utilisateur doit rendre observable l'économie réalisée alors que pour le second utilisateur, la température doit être rendue observable. Dans les deux cas, ces informations devront être présentées sous une forme compréhensible et adaptée à la situation. La qualité hédonique d'un système interactif est plus délicate à cerner. C'est l'objet de recherches actives en relation notamment avec l'informatique affective initiée par Rosalind Picard [PIC 97].

L'exemple du système de chauffage montre que les besoins fonctionnels de l'utilisateur, de même ses capacités cognitives et sensori-motrices et les ressources d'interaction ne sont ni immuables, ni totalement prévisibles. Si l'on pose comme principe que les requis de conformités fonctionnelle et interactionnelle doivent être satisfaits en toute circonstance, il convient dès lors de se pencher sur la question de l'adaptation des systèmes interactifs. À cet égard, la recherche en Interaction Homme-Machine a-t-elle des réponses à proposer ? Le domaine de l'Interaction Homme-Machine s'est organisé en deux courants de recherche complémentaires : l'un, centré sur l'invention de nouvelles techniques d'interaction, vise à résoudre des micro-problèmes d'interaction précis pour des situations d'usage précises, par exemple, le pointage de cible dans une scène graphique 2D [BLA 11]. C'est *l'interaction homme-machine « à pointe fine »*. Le second courant observe une approche holistique selon laquelle le système est envisagé comme un tout pour des usages dans le monde réel avec sa diversité et ses aléas. C'est *l'interaction homme-machine systémique*. Les méthodes de conception, les architectures logicielles de systèmes interactifs, les boîtes à outils et les générateurs d'IHM relèvent de l'Interaction Homme-Machine systémique. Le problème de l'adaptation peut tirer profit de ces deux courants. Illustrons notre propos avec l'exemple de l'objectif humain suivant : indiquer au système le choix d'une valeur parmi N possibilités.

L'apparition conjointe de la souris et des écrans à points a rendu possible la mise en œuvre du concept de menu graphique linéaire inventé pour servir d'extension à la mémoire à court terme : il est plus facile pour l'humain de reconnaître une valeur parmi N que de la retrouver ex nihilo [NOR 79]. S'appuyant sur la loi de Fitts (expliquée et exploitée dans [CAR 83]), on crée ensuite le menu circulaire qui permet de réduire le temps de sélection d'un élément sous réserve que le menu comporte au plus 8 éléments. Puis, les smart phones introduisent de nouvelles contraintes. Mais, grâce aux capteurs de mouvements, de nouvelles opportunités d'innovation sont offertes : le menu Wavelet permet d'afficher des listes hiérarchiques sur une petite surface [FRA 10] ou un menu linéaire se métamorphose en feuillets 3D lorsque l'utilisateur incline le téléphone (voir Figure 9.1). Sur les tables interactives multipoints, chaque doigt de la main vient servir de point d'entrée au menu MTM [BAI 08], comme les accords que l'on forme avec les touches d'un piano [BAU 10]. Si la projection d'informations a lieu sur un mur alors l'ombre du doigt, plutôt que le doigt, permet de désigner la cible hors de portée [SHO 07] ou encore un pico-projecteur sert à la déplacer vers soi pour la manipuler [CAO 06].

Ces exemples montrent que la recherche en IHM « à pointe fine » a su décliner le concept de menu au fur et à mesure de l'apparition de nouvelles ressources d'interaction mais sans se préoccuper de son intégration dans les systèmes. En complément, une approche systémique de l'IHM comme la recherche sur l'adaptation des interfaces homme-machine vise à produire des systèmes interactifs capables de faire appel à la forme de menu la plus appropriée à la situation en sorte que l'utilisateur atteigne ses objectifs (dans notre exemple, choisir une valeur parmi N) dans les meilleures conditions.



**Figure 9.1 :** Déclinaison du menu. De gauche à droite et de haut en bas : le menu Wavelet sur téléphone mobile [FRA 10], Menu polymorphe 2D/3D, MTM (Menu MultiTouch) pour table interactive multipoints [BAI 08], Shadow Reaching : sélection de cible avec son ombre [SHO 07].

#### 4 Titre de l'ouvrage

L'adaptation des IHM est donc un problème qui exige à la fois des connaissances pointues en Interaction Homme-Machine mais aussi des savoirs sur les systèmes interactifs dans leur ensemble, dépassant le domaine de l'Interaction Homme-Machine pour aller piocher dans de nombreux aspects de l'informatique, Génie Logiciel et systèmes répartis notamment. Si, comme nous le montrons dans ce chapitre, le Génie Logiciel et les systèmes répartis traitent de l'adaptation des logiciels, le facteur humain en est souvent exclu. C'est la raison pour laquelle, en 1999, nous avons introduit le concept de plasticité des interfaces homme-machine pour qualifier l'adaptation centrée utilisateur [THE 99]. C'est l'objet de ce chapitre que nous structurons en deux grandes parties : l'une pour définir avec précision le concept de plasticité des IHM et son espace problème (section 9.2), l'autre, en deux volets, consacrée à la mise en œuvre technique : les cadres de référence (sections 9.3 et 9.4) et nos recommandations de mise en œuvre (section 9.5). Nous concluons sur les acquis et les problèmes ouverts.

### 9.2. Plasticité : définitions et espace problème

La plasticité de l'interface homme-machine d'un système interactif dénote la capacité d'adaptation de cette interface au contexte d'usage pour en préserver l'utilité et l'utilisabilité [THE 99] et, par extension, la valeur [CAL 07] tout en accordant à l'utilisateur les moyens de contrôle adéquats [COU 06].

Avant de reprendre en détail les éléments clefs de notre définition (capacité d'adaptation en 9.2.2, contexte d'usage en 9.2.3, contrôle utilisateur en 9.2.4, et utilité-utilisabilité-valeur en 9.2.5), il convient de nous attarder en 9.2.1 sur une propriété voisine : l'*élasticité*.

#### 9.2.1. Plasticité et élasticité

Le domaine de l'informatique en nuage (« cloud computing ») utilise « élasticité » pour désigner la capacité du nuage à fournir des services évolutifs en sorte de répondre de manière ajustée, par exemple, à des demandes de montée en charge<sup>2</sup>. En sciences économiques, l'élasticité désigne la faculté d'adaptation d'un phénomène économique à des influences extérieures. En statistique, il s'agit du rapport des accroissements de deux variables. Ce concept est étudié également en physique, en biologie, et en d'autres domaines.

---

<sup>2</sup> Elasticité, définition du NIST : "capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time" (<http://csrc.nist.gov/groups/SNS/cloud-computing/>)

En résistance des matériaux, un corps solide élastique se déforme sous l'effet d'une charge mais, après décharge, il reprend intégralement sa forme initiale [FRA 92]. L'application d'une charge supérieure à la limite d'élasticité du solide provoque, soit une rupture brutale quasi sans déformation préalable (cas des matériaux fragiles), soit une déformation permanente dite déformation plastique (cas des matériaux ductiles) qui vient prolonger l'intégrité du matériau. Le système nerveux des mammifères, dont on dit qu'il est plastique, est capable de modifier son réseau de connexions synaptiques pour circonvenir une lésion.

Cette brève analyse de la terminologie formalisée dans des disciplines établies comme la physique, nous fait opter pour le terme « plasticité des IHM » qui perpétue la disponibilité au-delà des limites de l'élasticité, incluant même la faculté d'autoréparation.

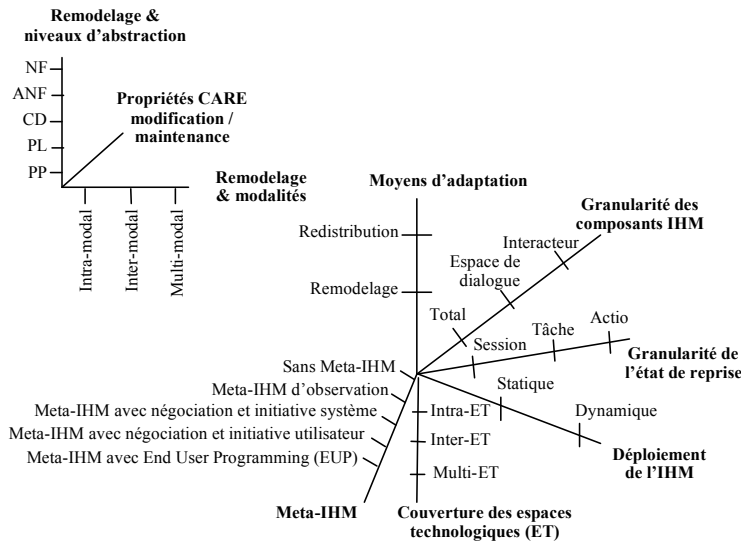


Figure 9.2 : Un espace problème pour la plasticité des IHM.

### 9.2.2. Capacité d'adaptation et son espace problème

L'adaptation des logiciels fait l'objet de recherches actives dans chacune des spécialités de l'informatique. Pour clarifier notre positionnement, nous proposons de caractériser la capacité d'adaptation d'une IHM dans l'espace multidimensionnel de la Figure 9.2. D'autres cadres sont tout aussi éligibles [CAL 07] [CAL 10]. L'espace problème ici retenu pour la plasticité des IHM comprend sept axes orthogonaux explicitant chacun une classe de questions qu'il convient de poser lorsque l'on conçoit ou que l'on analyse une IHM plastique.

#### 9.2.2.1. Moyens d'adaptation

L'IHM du système interactif est-elle adaptée par remodelage (c.-à-d. par changement de forme ou d'apparence perceptible pour l'utilisateur) ou bien par redistribution (c.-à-d. par migration vers d'autres ressources d'interaction) ou bien l'adaptation applique-t-elle une combinaison des deux leviers ?

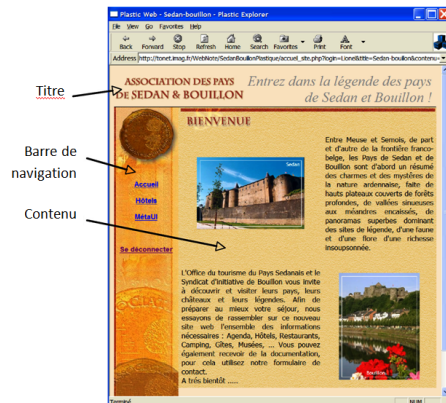


Figure 9.3 : Le site Sedan-Bouillon exploré avec un PC.

L'exemple de la version plastique du site de Sedan-Bouillon, site web de promotion des pays de Sedan et Bouillon, illustre l'exploitation conjointe du remodelage et de la redistribution. La Figure 9.3 montre la page d'accueil visualisée sur une station de travail de type PC. Elle comprend un titre, une barre de navigation et un contenu. L'utilisateur, qui explore le site à partir d'un PC, se connecte également au site via un PDA (Personal Digital Assistant). La plate-forme comprend maintenant deux nœuds : le PC et le PDA. Une redistribution d'IHM est proposée à

l'utilisateur (haut de la Figure 9.4) : celui-ci choisit d'avoir le titre et le contenu sur PC et souhaite disposer sur PDA du titre et de la barre de navigation. Le bas de la Figure 9.4 montre le résultat. Nous constatons que la barre de navigation, qui vient de migrer du PC vers le PDA, a également fait l'objet d'un remodelage pour tenir compte de la surface d'écran disponible sur le PDA. De même, sur le PDA, le titre contient l'essentiel : le nom du site. En revanche, le slogan « Entrez dans la légende des Pays de Sedan et Bouillon », qui est une information secondaire, a disparu.



**Figure 9.4 :** Le site Sedan-Bouillon en version distribuée entre les ressources d'interaction du PC et du PDA. En haut de la figure, le panneau de contrôle servant de méta-IHM de négociation en sorte que l'utilisateur exprime ses souhaits de redistribution de l'IHM de Sedan-Bouillon.

Il peut être utile de caractériser plus avant la nature d'un remodelage. Nous proposons pour cela le sous-espace tridimensionnel dessiné en partie gauche de la Figure 9.2. Le premier axe de l'espace du remodelage concerne les niveaux d'abstraction logiciels susceptibles d'être affectés par un remodelage. Cet axe caractérise l'impact de l'adaptation depuis des ajustements de surface de nature cosmétique jusqu'à une réorganisation profonde du système interactif. Nous utilisons pour cela les cinq niveaux d'abstraction du modèle d'architecture Arch [ARC 92] : par ordre croissant, la Présentation Physique (PP), la Présentation

Logique (PL), le Contrôleur de Dialogue (CD), l'Adaptateur de Noyau Fonctionnel (ANF) et le Noyau Fonctionnel (NF)<sup>3</sup> :

- remodelage de niveau Présentation Physique (PP). Le remodelage consiste en un changement d'apparence et/ou de disposition relative des interacteurs, mais conserve les classes d'interacteurs. C'est le cas de la barre de navigation de l'exemple Sedan-Bouillon ou du menu polymorphe 2D/3D de la figure 9.1.

- remodelage de niveau Présentation Logique (PL). Le remodelage consiste à remplacer des interacteurs par d'autres fonctionnellement équivalents. Par exemple, dans une IHM graphique, le remplacement d'un menu graphique par un champ de saisie est un remodelage de niveau PL. Ces deux interacteurs sont fonctionnellement équivalents (ils permettent tous deux de spécifier une valeur), mais ils n'ont pas les mêmes propriétés interactionnelles, tant du point de vue cognitif que du point de vue articulatoire : on l'a vu plus haut, pour l'utilisateur, générer est plus coûteux que reconnaître et, de manière générale, saisir est davantage source d'erreurs que sélectionner.

- remodelage de niveau Contrôleur de Dialogue (CD). Il s'agit d'une réorganisation de l'ordonnement des tâches. Par exemple, pour la tâche d'achat d'un billet de train, des sous-tâches qui pouvaient être exécutées de manière alternée lorsque l'achat est effectué au domicile, deviennent séquentielles pour améliorer le guidage lorsque cet achat est effectué depuis une borne en gare, contexte réputé stressant.

- remodelage de niveau Adaptateur de Noyau Fonctionnel ou de niveau Noyau Fonctionnel. Le remodelage se manifeste par l'ajout ou le retrait de services métier. Par exemple, pour un système de gestion du chauffage, on aura jugé inopportun de programmer sa maison à distance sur un téléphone mobile alors que cette possibilité est indispensable depuis un PC ou un dispositif dédié du domicile. Dès lors, sur téléphone mobile, l'accès aux commandes de programmation ne sera pas possible. Ou bien, une étude préalable aura identifié que la programmation devra être accessible en toute circonstance, mais que cette nécessité sera peu fréquente. Dans ce cas, les fonctions de programmation seront également accessibles depuis le téléphone, mais pas de manière prégnante comme sur le PC ou comme sur le dispositif dédié. On aurait alors recours à un remodelage de niveau CD. La Figure

---

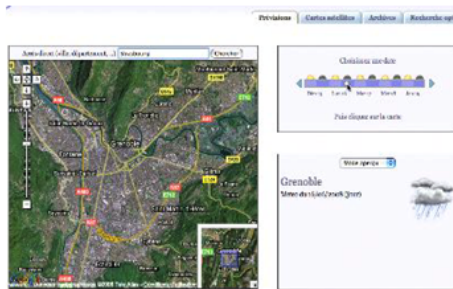
<sup>3</sup> Pour rappel, le NF implémente les services métier (ou services dépendants du domaine d'application) du système interactif. A l'autre extrême, PP est la « couche » réalisée avec les widgets (appelés aussi interacteurs) de boîtes à outils disponibles sur la plate-forme (e.g., AWT, SWING). En clé de voûte, CD est l'ordonnanceur des tâches et donc des appels aux services du NF (par exemple, le CD d'un éditeur de document impose qu'il faut ouvrir un document avant de l'éditer). ANF et PL servent d'intermédiaires correctifs pour minimiser toutes sortes de dépendances entre, d'une part, le monde purement NF et le monde purement IHM, et d'autre part, entre le CD (orienté buts et services) et le monde de rendu concret PP. Grâce au niveau PL, une IHM graphique peut être remplacée par une IHM vocale sans modifier le CD.



9.5 montre un exemple de remodelage au niveau ANF de l'IHM d'un service météorologique accessible depuis un PC et un Nabaztag.

De manière générale, notons qu'un remodelage de niveau d'abstraction donné a un effet de bord sur le niveau d'abstraction inférieur. Autrement dit, un remodelage de niveau NF ou ANF implique un remodelage de niveau CD et, par cascade, un remodelage des niveaux PL et PP.

a) IHM météo sur PC



b) IHM météo sur Nabaztag



**Figure 9.5 :** Remodelage de niveau Adaptateur de Noyau Fonctionnel (ANF) de l'IHM du service météorologique. Alors que la version PC offre de nombreuses fonctions, le Nabaztag ne renseigne que sur le temps qu'il fait en un lieu donné (choisi au préalable par l'utilisateur via le PC). Ces renseignements sont exprimés sous forme de trois LED dont les couleurs et le rythme de clignotement traduisent le temps (par exemple, trois LED jaunes clignotant gaiement signifient qu'il fait beau). Cette modalité graphique est complétée par une expression vocale accompagnée de la valeur de la température « Aujourd'hui, Soleil, 20 degrés ! » annoncée aux heures choisies au préalable par l'utilisateur via le PC.

Alors que l'axe « niveau d'abstraction » de l'espace du remodelage mesure l'impact d'un remodelage sur le logiciel, le deuxième axe, avec le concept de modalités d'interaction<sup>4</sup>, caractérise l'impact du remodelage perceptible pour l'utilisateur : Alors que l'axe « niveau d'abstraction » de l'espace du remodelage mesure l'impact d'un remodelage sur le logiciel, le deuxième axe, avec le concept de

<sup>4</sup> Une modalité d'interaction d'entrée ou de sortie se définit par le couple <langage de représentation, ressource d'interaction> noté <l, r> au sens de [NIG 94] [NIG 96]. Par exemple, <LN, microphone> et <LN, clavier> désignent deux modalités d'entrée en langue naturelle modélisée par la grammaire LN. Dans le cas de la première modalité d'entrée, les phrases nécessitent un microphone tandis que pour la seconde, les phrases sont saisies au moyen d'un clavier.

modalités d'interaction, caractérise l'impact du remodelage perceptible pour l'utilisateur :

- le remodelage est intramodal (par exemple graphique vers graphique) lorsque la modalité sensorielle humaine est préservée (la ressource d'interaction est préservée, mais le langage de représentation des informations change).

- le remodelage est intermodal (on dit aussi transmodal) lorsqu'il y a substitution de modalité par une autre modalité (le langage de représentation et la ressource d'interaction changent comme dans le remplacement d'interacteurs graphiques par des interacteurs vocaux).

- le remodelage est multimodal dès lors qu'il y a changement dans les combinaisons de modalités selon les propriétés CARE (Complémentarité, Assignation, Redondance, Equivalence) [COU 95] qui caractérisent la manière dont les modalités d'interaction sont exploitées. Un remodelage peut soit conserver, soit modifier, ces propriétés. Par exemple, parce que la puissance de calcul disponible chute, une complémentarité synergique peut devenir une complémentarité alternée : l'énoncé vocal « mets ça là » accompagné simultanément des gestes déictiques pour désigner le « ça » et le « là » est multimodal synergique (geste et parole sont produits en même temps). Cet énoncé devient multimodal alterné dès lors que l'utilisateur doit utiliser les différentes modalités en séquence (énoncé vocal « mets ça » suivi du geste pour désigner le « ça », puis énoncé vocal « là » suivi du geste pour désigner le « là »). Notons qu'un système monomodal peut devenir multimodal et vice versa. C'est le cas de la météo de la Figure 9.5 : on passe d'une IHM monomodale graphique en sortie sur PC vers une IHM multimodale graphique+son sur Nabaztag. De plus, la multimodalité du Nabaztag est redondante en sortie pour ce qui est de l'expression du temps (graphique LED + énoncé vocal – trois LED jaunes clignotantes + énoncé vocal « soleil ! ») alors qu'elle est complémentaire pour l'expression de la température (énoncé vocal seulement de la valeur de la température).

#### 9.2.2.2. Granularité des composants IHM

Les moyens d'adaptation (remodelage et redistribution) s'appliquent-ils à toute l'IHM prise comme une unité logicielle ou bien l'unité d'adaptation est-elle un sous-ensemble de cette IHM depuis l'espace de dialogue (par exemple, une fenêtre ou un canevas) jusqu'à un simple interacteur (par exemple, un menu) ? Un espace de dialogue (on dit aussi espace de travail) est un groupement d'informations nécessaires et suffisantes pour l'accomplissement d'une tâche. Une tâche pouvant être composée de sous-tâches, un espace de dialogue peut être composé de sous-espaces de dialogue. L'ensemble des espaces de dialogue d'un système interactif et leurs enchaînements constituent son IHM Abstraite. Nous reviendrons sur ce concept en section 9.3.

Le scénario-type du projet Aura illustre une adaptation de grain total [SOU 03] [SOU 05] : l'utilisateur édite un texte chez lui avec Word. Il se rend ensuite au bureau, laissant son travail en chantier. À son arrivée au bureau (déectée automatiquement par le système), Aura lui propose de poursuivre sa tâche d'édition du document avec les ressources disponibles dans ce nouvel environnement (Unix et Emacs). Du point de vue de l'utilisateur, l'IHM de l'éditeur de texte a migré de l'ordinateur personnel du domicile vers la station de travail professionnelle et a été entièrement remodelée pour s'adapter à la nouvelle plate-forme. Comme, à ce niveau de granularité, le système interactif est manipulé comme une entité indivisible, la redistribution n'est possible que sous la forme d'une migration totale d'une plate-forme vers une autre.

Dans l'exemple de Sedan-Bouillon, la granularité d'adaptation (par migration et remodelage) est celle de l'espace de dialogue : la barre de navigation, le titre, les requêtes et les résultats constituent des espaces de dialogue. Au grain le plus fin, l'adaptation se fait à l'échelle d'un interacteur. Par exemple, dans FlexClock, qui répond aux objectifs humains de déterminer l'heure et la date [GRO 02], lorsque la surface de l'écran disponible n'est plus assez grande pour afficher l'heure sous la forme d'un cadran à aiguilles, l'interacteur cadran est remplacé par un interacteur capable d'afficher l'heure de manière textuelle. S'il n'y a plus assez de place pour afficher l'heure et la date, l'espace de dialogue « date », qui correspond à une tâche jugée moins prioritaire, est supprimé.

#### 9.2.2.3. Granularité de l'état de reprise

Que l'adaptation se fasse par remodelage et/ou par redistribution, et quelle que soit la granularité des composants IHM affectés par cette adaptation, quel est, du point de vue de l'utilisateur, le grain de reprise ? En d'autres termes, la granularité de l'état de reprise mesure la continuité de l'interaction avec le système (et donc l'effort laissé à la charge de l'utilisateur) entre le moment qui précède l'adaptation et celui qui la suit pour atteindre ses objectifs.

L'utilisateur se retrouve-t-il en tout début de session ? Dans les sites web, il peut arriver qu'une requête de changement de langue nous repositionne sur la page d'accueil. On assiste alors à une reprise de niveau session. L'utilisateur doit-il recommencer, non pas depuis le début de session, mais seulement depuis le début de la tâche interrompue par le processus d'adaptation, ou bien peut-il reprendre sa tâche au point où il l'avait laissée avant l'adaptation (reprise au niveau action) ? Dans le cas de Sedan-Bouillon, la reprise a lieu au niveau tâche : si l'utilisateur était en train de renseigner un champ texte (par exemple une adresse d'hôtel) qu'il n'aurait pas validé avant l'adaptation, il doit ressaisir ce champ une fois l'adaptation réalisée. Dans Aura, l'utilisateur retrouve son document ouvert dans l'état où il l'avait laissé (y compris le point d'insertion courant). Le grain de reprise est alors celui de l'action physique.

#### 9.2.2.4. Déploiement de l'IHM

Quand l'adaptation a-t-elle lieu ? À la conception ou à l'installation du système (déploiement statique) ? À l'exécution (déploiement dynamique) ? Le niveau statique concerne les adaptations réalisées alors que le système interactif n'est pas en cours d'utilisation. Cela regroupe les niveaux « d'adaptation au développement » et « d'adaptation à l'installation » de la taxonomie de [THE 01], ainsi que les niveaux « avant la première utilisation » et « entre sessions » de celle de [DIE 93].

Le niveau dynamique caractérise les systèmes interactifs capables d'adaptation au moment de leur exécution, au cours de l'interaction avec l'utilisateur. On peut affiner cet axe en distinguant les adaptations dynamiques pré-calculées à la conception et les adaptations calculées dynamiquement. La première classe d'adaptation dynamique permet d'adapter un système interactif à l'exécution pour un ensemble de contextes d'usage identifiés au moment de la conception. La deuxième classe d'adaptation dynamique est celle des systèmes capables d'adaptation à un contexte d'usage qui n'a pas été prévu par les concepteurs. Par exemple, les sites web tendent à proposer une version « bureau » et une version « mobile »<sup>5</sup> : c'est une illustration de l'adaptation pour un ensemble de contexte prévus à la conception. À l'inverse, des systèmes comme SUPPLE [GAJ 04] ou Arnould [GAJ 05][GAJ 08] permettent de générer l'IHM en fonction des dimensions de l'écran et de l'utilisation passée par l'utilisateur : c'est une illustration de l'adaptation à des contextes non explicitement prévus à la conception.

#### 9.2.2.5. Couverture des espaces technologiques

Quel degré d'hétérogénéité technologique l'IHM est-elle capable de supporter ? Un espace technologique est un « contexte de travail [technique] comprenant un ensemble de concepts reliés, de connaissances, d'outils, de compétences requises et de possibilités<sup>6</sup> » [KUR 02]. Par exemple, les mondes Java et C# constituent deux espaces technologiques distincts. La plupart des IHM sont implémentées au sein d'un unique espace technologique comme Tcl/Tk, Swing, HTML. Cette homogénéité ne tient plus pour les IHM plastiques dont la distribution simultanée sur différents dispositifs peut requérir de traverser des espaces technologiques.

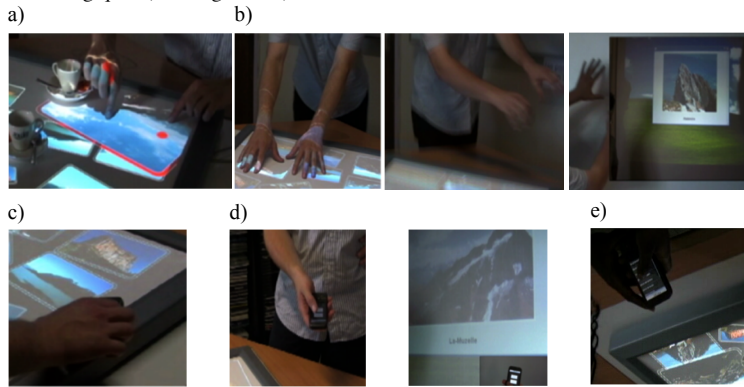
Dès lors, les possibilités sont les suivantes : l'adaptation est-elle intra-espace technologique (par exemple, de Java vers Java), inter-espaces technologiques (par exemple, de Java vers WML) ou multi-espaces technologiques combinant par

---

<sup>5</sup> A titre d'illustration, le journal Le Monde propose un site dédié à l'affichage sur station de bureau (<http://www.lemonde.fr>) et un autre dédié à l'affichage sur téléphone mobile (<http://mobile.lemonde.fr/>)

<sup>6</sup> "A technological space is a working context with a set of associated concepts, body of knowledge, tools, required skills, and possibilities." [KUR 02]

exemple Java et HTML avant ou après adaptation. Dans Sedan-Bouillon, l'adaptation est intra-espace technologique (de HTML à HTML). Photo-Browser, un album de photos numérique plastique, illustre le cas de l'adaptation multi-espaces technologiques (voir Figure 9.6).



**Figure 9.6 :** Exemple de plasticité multi-espaces technologiques. L'IHM de Photo-Browser est une combinaison dynamique d'espaces technologiques Java, HTML et Tcl/tk. a) Seule, la table est présente : IHM centralisée ; espace monotecnologique Tcl/Tk ; tâche de tri de photos mono- ou multi-utilisateurs. b) Méta-IHM de commande de redistribution par un geste de balayage des deux mains de la table vers le mur. Table et PC présents : IHM distribuée sur table et mur ; espace multitecnologique Tcl/tk, HTML ; tâches de tri de photos et partage de la photo actuellement sélectionnée sur la table. c) Méta-IHM de connexion du gPhone à la plate-forme de Photo-Browser par pose du gPhone sur la table. d) Table, PC et gPhone présents : IHM distribuée sur table, mur et téléphone ; espace multitecnologique Tcl/tk, HTML, Java ; tâches de tri de photos, de partage public de la photo au centre d'intérêt, changement du centre d'intérêt depuis la table mais aussi au moyen du téléphone avec les boutons « next » et « previous ». e) le gPhone comme télécommande de sélection de la photo courante à partir d'une liste de noms. **NB.** Dans la version présentée ci-dessus, la reconnaissance des gestes est réalisée par Magicien d'Oz.

Photo-Browser permet de parcourir un album de photos de manière centralisée ou de manière distribuée en fonction des ressources d'interaction en présence et de l'environnement physique et social (un lieu public ou chez soi). Les ressources d'interaction possibles incluent une surface interactive multipoints Diamond Touch, un mur géré par un PC sur lequel est projetée la photo sélectionnée sur la table, et un Smart Phone Android. L'IHM de PhotoBrowser est une composition (dynamique) multi-espaces technologiques comprenant un composant Tcl-Tk exécutable sur la

table, un navigateur web (HTML) pour feuilleter les images une par une sur le mur, et, sur le gPhone, deux composants Java : d'une part, le menu des noms de toutes les images de l'album, d'autre part les boutons « Next » et « Previous » pour parcourir les photos en séquence. Dans les deux cas, le gPhone joue le rôle de télécommande. L'apparition et la disparition des ressources d'interaction sont commandées par une méta-IHM (concept présenté en 9.2.4).

Les deux derniers concepts clefs de la plasticité des IHM, contexte d'usage et méta-IHM, sont présentés dans les deux sous-sections suivantes.

### 9.2.3. Contexte d'usage

L'importance du contexte d'usage est reconnue depuis longtemps en conception des systèmes interactifs. La conception contextuelle (contextual design) de Beyer et Holtzblatt, fondée sur la collecte et l'organisation de données de terrain (contextual inquiry), en est un exemple type [BEY 98]. Les modèles de l'Action Située (Situated Action) [SUC 87], la Théorie de l'Activité [BAR 97], la Cognition Distribuée (Distributed Cognition) [HAL 94] ou encore la conception par scénarios [CAR 00] visent autre chose que la décomposition stricte de GOMS [CAR 83] en tâches et sous-tâches en considérant le contexte comme élément directeur.

Si le contexte est une notion admise depuis longtemps en conception des systèmes interactifs, en pratique il n'est utilisé qu'en phase amont du processus de développement. Tout ce qui relève du contexte se dilue progressivement au cours du processus de développement et le triangle classique en IHM « utilisateur-tâche-machine » ne fonctionne que pour un contexte statique prévu à l'avance. Avec l'émergence de l'intelligence ambiante, le contexte est redécouvert et placé au cœur des débats sans pour autant faire l'objet d'une définition consensuelle claire et définitive.

Toutefois, l'analyse de l'état de l'art [COU 05] conduit à ce double constat : (1) il n'y a pas de contexte sans contexte [BRE 02]. Autrement dit, le contexte n'existe pas en tant que tel. Il émerge, ou se définit, pour une finalité précise. (2) Le contexte est un ensemble d'informations structurées, partagé et évolutif qui sert l'interprétation [WIN 01]. La nature des informations, de même que l'interprétation qui en est faite, dépendent de la finalité. La finalité qui nous concerne ici est l'adaptation logicielle des systèmes ambiants. Parmi les nombreuses ontologies qui portent sur le contexte avec l'adaptation logicielle pour finalité (cf. AMIGO [EUZ 08] et bien d'autres [DEY 01]), celle de Crowley et al. [CRO 02, REY 05, REI 07] présente, selon nous, un bon pouvoir de généralité. Nous la présentons ci-dessous dans ses grandes lignes. Nous verrons ensuite comment elle s'applique à la modélisation du contexte d'usage.

### 9.2.3.1. *Ontologie pour la notion de contexte selon Crowley et al.*

Un contexte est défini sur un ensemble E d'entités, un ensemble Ro de rôles que certaines entités de E sont capables d'assurer et un ensemble Rel de relations entre les entités de E. Les entités, rôles et relations sont modélisés par des expressions d'observables. Un observable est une information captée et/ou calculée par le système. Par exemple, si l'on considère une salle de réunion, E dénote les participants, Ro comprend les rôles d'orateur et d'auditeur et Rel peut inclure des relations spatiales comme « l'entité e1 (qui joue le rôle d'orateur) est en face de e2 (qui joue le rôle d'auditeur) ».

Un contexte est un réseau de situations où chaque nœud représente une situation et où les arcs dénotent les conditions de transition entre situations. Les conditions de passage entre deux situations sont les suivantes :

- Modification de la cardinalité de E. Dans l'exemple de salle de réunion, deux retardataires entrent dans la salle et sont reconnus par le système en tant que participants (leurs observables correspondent aux caractéristiques et au comportement de participant). On change de situation. Le système y répond, par exemple, en leur fournissant un résumé des échanges depuis le début de la réunion.

- Changement d'association « rôle-entité ». Dans notre cas, l'orateur cède son rôle au profit d'un autre participant. En réponse à ce changement de situation, le système oriente la caméra d'enregistrement de la réunion en direction du nouvel orateur et « branche » le portable du nouvel orateur au seul vidéoprojecteur de la salle. Le système sait que chaque participant a sa présentation sur son portable personnel.

- Changement de relations entre entités. Par exemple, le participant e, qui était devant e', est maintenant derrière e'. Sur ce changement de situation, le système permet à e de visualiser sur son propre ordinateur les diapositives vidéo projetées en public. Le système sait que e est maintenant trop loin pour bien voir les diapositives du groupe de travail.

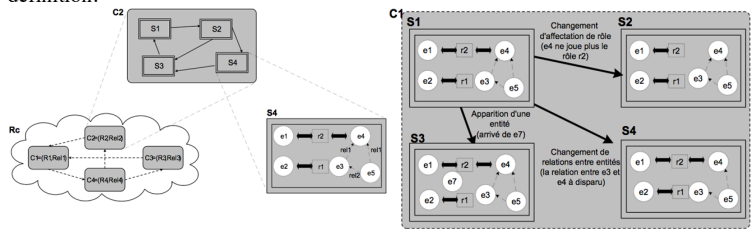
Ces définitions de contexte et de condition de changement de situation impliquent que toutes les situations d'un même contexte partagent les mêmes ensembles E, Ro, Rel. Il suffit que l'une des conditions suivantes soit satisfaite pour qu'il y ait changement de contexte :

- E est remplacé, tout ou partie, par un autre ensemble d'entités. Par exemple, E, l'ensemble des participants de la réunion où la notion de « participant » est caractérisée par un ensemble d'observables, est maintenant l'ensemble des membres d'une famille où « membre de famille » est caractérisé par un autre ensemble d'observables, l'intersection entre ces deux ensembles n'étant pas nécessairement vide.

- Ro est remplacé, tout ou partie, par un autre ensemble de rôles. Dans l'exemple de la salle de réunion, si le rôle d'orateur n'est plus rempli, on change de contexte (celui de discussions informelles, par exemple).

- Rel est remplacé, tout ou partie, par un autre ensemble de relations. Pour la salle de réunion, en plus des relations spatiales, des relations temporelles peuvent maintenant devenir pertinentes.

Au bilan, le contexte est modélisé à deux niveaux d'affinement : un graphe de contextes où chaque nœud dénote un contexte – chaque contexte étant défini par un triplet unique de trois ensembles (E, Ro, Rel) et dont les arcs correspondent à une condition de changement de contexte. À son tour, chaque nœud du graphe de contextes s'affine en un réseau de situations où chaque nœud dénote une situation – chaque situation étant définie par des instances sur E, Ro, Rel, et dont chaque arc dénote une condition de changement de situation. La Figure 9.7 traduit cette définition.



**Figure 9.7 :** Modèle de contexte (d'après [REY 05]). À gauche : Contexte comme graphe de contextes (Rc pour « Réseau de contextes » comprenant ici quatre contextes C1, C2, C3, C4). Chaque contexte Ci est un graphe de situations. Les situations d'un contexte donné partagent le même ensemble d'Entités, de Relations et de Rôles. À droite, les conditions de changements de situations au sein d'un contexte.

L'ontologie que nous venons de présenter n'impose en aucune façon la nature des entités, des rôles, des relations et des observables. Il s'agit de classes abstraites dont il convient de définir les spécialisations pour chaque exploitation cible. Comment l'appliquer au contexte d'usage ?

### 9.2.3.2. Contexte d'usage comme spécialisation de l'ontologie de Crowley et al.

Les entités d'un contexte d'usage vu comme une spécialisation de l'ontologie de Crowley et al. sont organisées en trois classes : l'utilisateur, la plate-forme, l'environnement physique et social.



L'utilisateur est représenté par des attributs (c.-à-d. des observables) et des fonctions (c.-à-d. des rôles) qui décrivent la personne archétype censée utiliser le système et/ou la personne en train d'utiliser le système. Ces informations vont du profil-type (compétences métier et informatique, âge, taille, capacités sensori-motrices, etc.), aux préférences et habitudes inférées par le système, voire l'activité et la tâche en cours au sein de cette activité. Par exemple, dans SUPPLE [GAJ 04][GAJ 08], l'utilisateur est modélisé par ses performances en terme de la loi de Fitts et par les traces de ses utilisations précédentes du système interactif.

L'environnement est représenté par des attributs (c.-à-d. des observables) et des fonctions (rôles) qui décrivent l'espace spatio-temporel cible où le système sera utilisé et/ou est actuellement utilisé. Comme pour le modèle de l'utilisateur, le nombre de dimensions pertinentes est grand. Cela peut inclure un observable correspondant à une localisation spatiale numérique (par exemple, les coordonnées géographiques fournies par un GPS) ou à une localisation spatiale symbolique (chez soi, à la gare, dans la rue, dans un train, en voiture, à Rome, dans second life !) ou encore à une région plutôt qu'à un point dans un système de représentation spatiale (« entre Grenoble et Nice ») ; de même, un observable correspondant à une localisation temporelle numérique ou symbolique (« il est 14h, jeudi 2 février 2011 » ou bien « en début d'après midi »), mais aussi une région temporelle (« entre midi et 16h00 »). Le modèle de l'environnement peut inclure les règles sociales en usage dans cet environnement, de même ses conditions physiques comme le niveau de luminosité, la température, la qualité de l'air et bien d'autres, elles aussi exprimées de manière numérique et/ou symbolique.

La plate-forme décrit les ressources de calcul, de communication, d'interaction, de même que les ressources logicielles de base qui lient l'environnement physique au monde numérique. On entend par « logiciel de base » le système d'exploitation, ses pilotes (drivers) et plus généralement, l'environnement d'exécution (bibliothèques, interpréteurs, intergiciel, etc.). Dans le monde interactif conventionnel, la plate-forme est élémentaire : elle regroupe en un tout quasi immuable « ressources de calcul, ressources de communication, ressources d'interaction, logiciel de base ». Avec l'intelligence ambiante, l'utilisateur devient son propre constructeur d'écosystèmes (cf. exemples du chapitre d'introduction de cet ouvrage). Plus généralement, n'importe quel objet du quotidien peut jouer le rôle de ressources d'interaction (y compris la surface de la peau comme pour Skininput [HAR 10]) et faire dynamiquement partie ou être retiré de la plate-forme. Alors que les travaux de recherche sur la plasticité des IHM se sont concentrés sur l'adaptation à la plate-forme, à notre connaissance, il n'existe pas de modèle de plate-forme assez riche pour inclure des clusters dynamiques hétérogènes. Dans les exemples cités ci-dessus (Sedan-Bouillon, CamNote, Photo-Browser), le modèle de la plate-forme est implicite, c'est-à-dire dilué dans le code des applications.

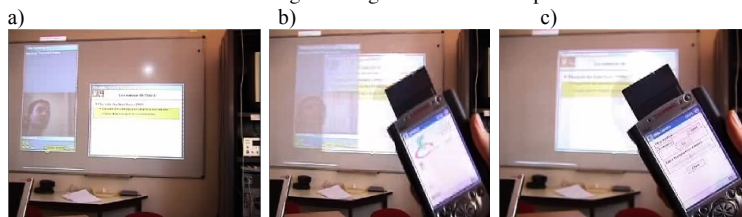
Nous venons de donner des exemples d'observables et de rôles pour les trois classes d'entités considérées comme clefs dans un contexte d'usage : l'utilisateur, la plate-forme et l'environnement physique et social. Les observables et les rôles doivent être complétés par l'ensemble des relations pertinentes que le système est censé assurer : par exemple, « l'utilisateur est ici », « ce dispositif est utilisé par tel utilisateur » ou « se trouve près de la porte », « il est bientôt l'heure de partir », etc.

Si le système interactif est censé s'adapter aux changements du contexte d'usage, l'utilisateur a parfois son mot à dire. Dans ce cas, comment ? En réponse, nous proposons le concept de méta-IHM.

#### 9.2.4. Méta-IHM et contrôle utilisateur

Une méta-IHM est un système interactif qui regroupe l'ensemble des fonctions (avec leur interface utilisateur) nécessaires et suffisantes pour évaluer et contrôler non seulement l'état mais aussi le comportement de systèmes ambiants et notamment leur processus d'adaptation [COU 06]. Cet ensemble est méta- parce qu'il sert d'arche générale à tout l'espace ambiant. Il donne à l'utilisateur final les poignées pour intervenir aussi bien sur le comportement fonctionnel de l'espace ambiant que sur son comportement interactionnel. Notons que le desktop est la méta-IHM des stations de travail conventionnelles.

Comme le montre la **Figure 9.2**, on distingue les méta-IHM d'observation, les méta-IHM de négociation (avec initiative du système ou de l'utilisateur) et les méta-IHM relevant du « End-user Programming/End-user Development ».



**Figure 9.8** : Visualisateur de diapositives CamNote [DEM 05]. Exemple de méta-IHM d'observation qui rend explicite la migration du panneau de contrôle entre le PC et le PDA. a) IHM centralisée sur PC avant l'arrivée du PDA. b) Méta-IHM d'observation. Le panneau de contrôle visible sur le mur glisse vers la diapositive pour s'y fondre progressivement. c) Le panneau de contrôle a disparu du mur. Il réapparaît sur le PDA après avoir été remodelé sous forme de gros boutons pour faciliter la sélection en situation de mobilité.

#### 9.2.4.1. Méta-IHM d'observation

Une méta-IHM d'observation rend observable l'état du processus d'adaptation mais l'utilisateur ne peut intervenir sur ce processus sauf, au mieux, le suspendre pour un éventuel retour arrière. Par exemple, le menu polymorphe 2D/3D de la Figure 9.1 change progressivement de forme, en interaction fortement couplée<sup>7</sup> avec l'inclinaison du Smartphone, permettant ainsi un aller-retour continu (mais interruptible et donc réversible) entre les deux formes stables 2D et 3D du menu. Les animations, qui déportent au niveau du processeur visuel humain l'évaluation du changement d'état d'une interface homme-machine [ROB 91], font aujourd'hui partie de l'appareillage de base des boîtes à outils de développement d'interfaces homme-machine centralisées. La question est ouverte lorsqu'il s'agit de traduire la migration de parties d'IHM dans l'espace. Dans l'exemple du présentateur de diapositives CamNote, la migration du panneau de contrôle du PC vers le PDA se traduit par la disparition du panneau qui vient se fondre progressivement dans la diapositive qui, elle, reste sur le PC pour réapparaître (par remodelage intramodal) sur le PDA (Figure 9.8).

#### 9.2.4.2. Méta-IHM avec négociation

Une méta-IHM avec négociation, que ce soit à l'initiative du système ou de l'utilisateur, permet à l'utilisateur de rester dans la boucle pour contrôler le processus d'adaptation. Par exemple, la méta-IHM de Sedan Bouillon présente sous forme d'un panneau de contrôle (Figure 9.4) les ressources d'interaction disponibles ainsi que les espaces de dialogue migrables. Elle permet à l'utilisateur de choisir à la volée la redistribution des espaces de dialogue du système sur l'ensemble des ressources d'interaction (dans ce cas, celles du PC et du PDA). Ou encore, dans Photo-Browser, le geste de balayage de la Figure 9.6 permet de faire migrer sur le mur la photo sélectionnée sur la table. La pose du gPhone sur le rebord de la table exprime le désir de faire intervenir le gPhone comme dispositif additionnel de la plate-forme. Ces gestes constituent la méta-IHM de Photo-Browser. Dans Sedan-Bouillon, le système prend l'initiative de l'adaptation (provoquée par la détection de l'arrivée du PDA) : il propose les possibilités de redistribution et l'utilisateur dispose. Dans le cas de Photo-Browser, l'utilisateur a l'initiative de l'adaptation. Concernant les rendus sur le gPhone utilisé comme télécommande, le choix entre les boutons de navigation séquentielle et le menu des noms de photos peut être laissé à l'utilisateur (via une méta-IHM de négociation) ou peut être décidé par le système en fonction de la situation (absence de méta-IHM). Mais, pour Photo-Browser

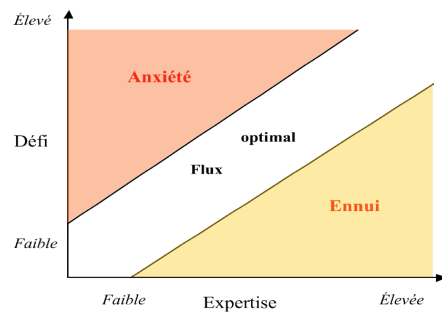
<sup>7</sup> Une interaction est fortement couplée (IFC) sur un intervalle de temps donné lorsque les systèmes humain et artificiel sont engagés de manière continue dans l'accomplissement d'actions physiques mutuellement observables et dépendantes sur cet intervalle. Le déplacement d'un objet graphique avec la souris relève de l'IFC [BER 99].

comme pour Sedan-Bouillon et Arnault [GAJ 05], ces méta-IHM de négociation ont été prévues à la conception et donc préprogrammées par des développeurs.

La prochaine étape, riche en défis, consiste à donner les pleins pouvoirs à l'utilisateur final en lui fournissant un véritable environnement de développement. C'est ce qu'on appelle le « end-user programming/end-user development » (EUP/EUD).

#### 9.2.4.3. End-user programming/end-user development (EUP/EUD)

Le EUP/EUD a comme objectif la démocratisation de l'informatique : il s'agit de permettre à des non-spécialistes de l'informatique de construire et de façonner leurs systèmes interactifs, de les mettre au point, les réutiliser et les maintenir ; tout cela sans compétence en Génie Logiciel (GL), ni motivation pour apprendre les « bonnes pratiques » du GL. Le défi posé par les EUP/EUD (qui vaut pour tout outil de développement logiciel) revient à établir le juste compromis entre le pouvoir d'expression de l'outil et le coût d'apprentissage pour l'utilisateur. D'après les psychologues, les conditions favorables à l'apprentissage tiennent au bon équilibre entre le défi posé à l'apprenant et son niveau d'expertise [REP 04]. Comme le montre la Figure 9.9, un trop grand défi par rapport à l'expertise de l'apprenant provoque anxiété et rejet. Inversement, l'absence de défi peut être génératrice d'ennui.



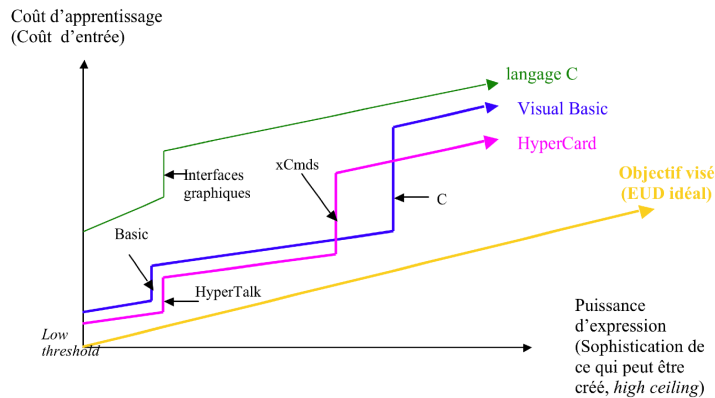
**Figure 9.9 :** Flux optimal d'apprentissage vu comme le rapport entre défi posé et expertise de l'apprenant (d'après [REP 04]).

Si l'on se réfère à la théorie de l'action de Norman [NOR 86], le rapport défi/expertise ne peut être optimal que si les distances d'exécution et d'évaluation que doit parcourir l'utilisateur en interaction avec un système sont conformes à ses compétences et motivations. Les tableurs (spreadsheets) offrent un bon exemple :

l'addition d'un ensemble de nombres s'obtient en appliquant l'opérateur SOMME à l'ensemble. Cette technique, qui fonctionne par analogie avec la compétence acquise dès l'école primaire, est facile à assimiler : les distances cognitives sont optimales. Par contraste, en langage impératif comme C, l'algorithme équivalent est loin des formes de pensée courantes :

```
sum = 0 ;
for (i=0 ; i<numItems ; i++) {
    sum += items [i] ;
}
return sum ;
```

Mais l'Homme est désireux de progrès. Par utilisation répétée de son EUP/EUD, il est amené à explorer de nouvelles possibilités sous réserve qu'il ne rencontre pas de barrières insurmontables pour son niveau d'expertise. D'où le profil idéal d'EUP/EUD à pente douce qu'illustre la Figure 9.10.



**Figure 9.10 :** *d'apprentissage en fonction de la puissance d'expression d'un EUP/EUD (d'après [MYE 00]).*

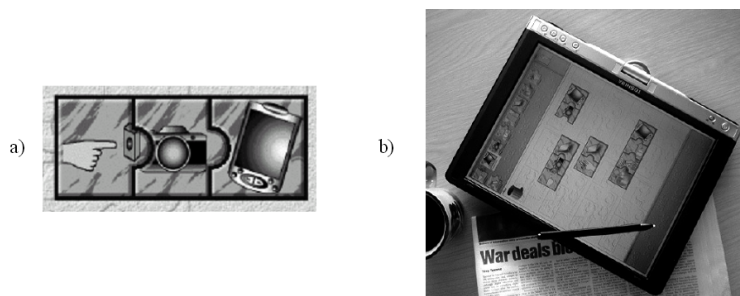
Par contraste avec le profil de l'EUP/EUD à pente douce et régulière :

- La courbe correspondant au langage C montre que la prise en main du langage est d'emblée difficile, et que, pour programmer des interfaces graphiques, l'utilisateur se heurte à un mur que représente la maîtrise de MFC (Microsoft Foundation Classes).

- Avec Visual Basic, la courbe commence plus bas. Le contact initial est moins violent. Mais la courbe présente deux montées verticales : lorsque l'utilisateur doit apprendre le langage Basic, puis lorsqu'il doit apprendre C.

- Hypercard est simple d'accès mais pose deux barrières : l'apprentissage du langage HyperTalk, puis celui de xCmds.

- La courbe en pente douce modélise le système idéal des EUP/EUD : un coût d'apprentissage et une puissance d'expression qui progressent régulièrement en cohérence avec le flux optimal d'apprentissage. Le droit d'entrée est quasi nul (low threshold) avec à termes, la capacité de produire des programmes/documents sophistiqués (high ceiling).

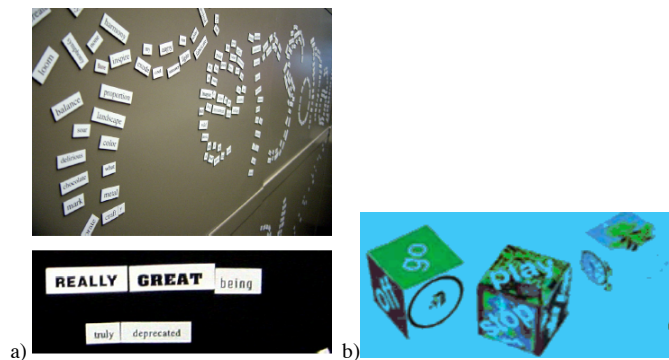


**Figure 9.11** : JigSaw [ROD 04] : langage de programmation graphique iconique selon la métaphore du puzzle. Les pièces désignent respectivement la sonnette, l'appareil photo et le PDA. Ce programme signifie que : « lorsque quelqu'un sonne à la porte, le prendre en photo et envoyer la photo sur mon PDA ».

En synthèse, l'utilisation d'une méta-IHM de type EUP/EUD doit se voir comme une expérience d'apprentissage. La conséquence pour la conception de telles méta-IHM est que l'utilisateur doit être capable de produire sans erreur des programmes simples pour des objectifs simples, puis, une fois la confiance acquise, l'outil doit lui permettre de découvrir progressivement des fonctions plus puissantes pour atteindre des objectifs complexes<sup>8</sup>. Les tableaux constituent l'exemple le plus réussi de EUP/EUD. Les mashups comme les Pipes de Yahoo! traduisent cette tendance avec les services et données du Web mais ils s'adressent encore à des utilisateurs très motivés. En matière de EUP/EUD pour l'intelligence ambiante, l'effort s'est

<sup>8</sup> C'est aussi ce que Carroll préconise avec les « training wheels », par analogie avec les stabilisateurs que l'on installe sur les vélos d'enfant [CAR 84]. Les stabilisateurs permettent à l'enfant de se familiariser avec la conduite en toute sécurité avant de passer au mode plus défiant sans roulettes.

jusqu'ici porté sur la syntaxe des langages auteur : syntaxe en langage pseudo-naturel comme dans CAMP [TRU 04], langage iconique graphique de JigSaw [ROD 04], programmation par l'exemple et par démonstration, etc. Avec Jigsaw, l'utilisateur programme son espace par assemblage de pièces de puzzle où chaque pièce représente un objet de l'environnement (Figure 9.11). Les mediaCubes relèvent du même principe de rapprochement avec le monde physique où chaque face de cube physique comporte un mot ou une icône [BLA 01]. L'utilisateur compose une phrase en assemblant et en orientant correctement les cubes (Figure 9.12). Signalons enfin Façade [STU 06] qui permet à l'utilisateur de reconstruire l'IHM de ses applications selon ses besoins en copiant/collant des interacteurs, voire en substituant des interacteurs par d'autres.



**Figure 9.12 :** Deux syntaxes concrètes pour construire des phrases en langage pseudonaturel. En a), les « poetic magnets » de CAMP [TRU 04] ; en b), les Media Cubes physiques de AutoHan porteurs de vocabulaire [BLA 01].

Nous avons jusqu'ici présenté trois des quatre composantes de la plasticité des IHM : capacité d'adaptation (9.2.2), contexte d'usage (9.2.3) et méta-IHM (9.2.4). Il nous reste à considérer la qualité de l'adaptation en termes d'utilité, d'utilisabilité et de valeur.

### 9.2.5. Utilité, utilisabilité, valeur

L'utilité (conformité fonctionnelle) et l'utilisabilité (conformité interactionnelle), présentées en introduction de ce chapitre, ont toujours servi de moteur à la recherche en Interaction Homme-Machine. Ces deux propriétés constituaient jusqu'ici les deux

critères de référence pour caractériser la valeur d'un système. Aujourd'hui, sont également considérées comme éléments de valeur des notions plus subtiles comme le plaisir (suscité par l'esthétique notamment), l'estime, l'affection ou l'attachement, mais aussi la conformité avec des principes de vie, culture, morale, éthique. Développons ce dernier point avant de revenir sur les fondamentaux du développement centré humain.

Respecter des valeurs éthiques signifie que le système ambiant doit être en mesure de modéliser les valeurs morales et légales de façon à prendre ou aider à prendre les « bonnes » décisions en cas de conflits. Actuellement, nous ne savons pas spécifier un comportement moral. L'approche simpliste consisterait à considérer les valeurs éthiques comme des contraintes supplémentaires que le système devrait satisfaire comme il en va pour toute contrainte. Encore faut-il que ces contraintes soient identifiées. Elles peuvent être très spécifiques (réduire la consommation d'énergie) ou très abstraites (ne jamais blesser un humain). Elles peuvent être très personnelles, mais aussi provenir du législateur ou de mécanismes régulateurs d'origines culturelles et sociales. Faut-il songer au développement de machines éthiques comme le suggère Allen [ALL 06] ? Nous ne répondrons pas à cette question : utilité et utilisabilité présentent, déjà à elles seules, leurs lots de difficultés.

Concernant l'utilité, de nombreuses méthodes d'analyse des besoins fonctionnels se sont penchées sur la question, tant en Interaction Homme-Machine qu'en Génie Logiciel avec des tentatives de rapprochement et d'intégration plus ou moins réussies. Ces méthodes et outils supports font cependant l'hypothèse que l'application est un système fermé. Or, l'émergence des solutions logicielles orientées services en concomitance avec la montée en puissance de la participation sociale médiée par les technologies [SHN 11] ouvre la voie à la construction, par l'utilisateur final, de son propre noyau fonctionnel. Par conséquent, l'utilité d'un système n'est plus totalement décidable a priori par les concepteurs. Autrement dit, la conception centrée utilisateur est en train d'échapper aux professionnels ce qui, au fond, est plutôt une bonne nouvelle. Il n'empêche que ce phénomène force à repenser la façon de développer des systèmes : ceux-ci doivent pouvoir être décomposés en petites unités ré-assemblables selon les besoins des utilisateurs. Cette tendance n'est pas sans rappeler la programmation orientée service, le défi supplémentaire étant ici de composer non plus seulement les noyaux fonctionnels des systèmes interactifs mais aussi leurs IHM.

Concernant l'utilisabilité, nous disposons de nombreux référentiels, parfois normalisés (par exemple l'ISO/IEC 9126), certains généralistes (pour exemples, des référentiels classiques tels que [SHA 91] [ABO 92] [BAS 93] [DIX 93] [NIE 94] [PRE 94] [IFI 96] [SHN 97] [CON 99] [VAN 99] [BRA 03] [STE 09], d'autres spécifiques (par exemple [MON 04] [NOG 08] pour le web ou [LOP 04] pour



l'adaptation). Par exemple, l'observabilité<sup>9</sup> est un critère d'utilisabilité général. Par opposition, la continuité de l'interaction [TRE 03], par exemple mesurée par le grain de reprise de la tâche qui permet de mesurer, en termes d'actions sur les dispositifs d'entrée, le coût de l'adaptation pour l'utilisateur, ou encore l'inter-utilisabilité<sup>10</sup> [KAR 05], appelée aussi utilisabilité horizontale [SEF 04], sont des critères d'utilisabilité propres à l'adaptation. Nous ne recommandons pas de référentiel particulier si ce n'est, dans l'espace du possible, se définir un ensemble de critères pertinents pour le sujet d'étude. Cet ensemble doit servir de référentiel systématique pour juger de la qualité plastique du système en question [GAR 11]. On trouvera en Annexe, un affinement de cette question.

### 9.2.6. Synthèse

Que retenir en quelques mots de cette section 9.2 sur le concept de plasticité ? La plasticité des IHM est une propriété multidimensionnelle complexe.

Plasticité des IHM signifie capacité d'adaptation, non pas à des changements de nature simple, mais au contexte d'usage pour lequel il n'existe pas de solution technique maîtrisée. Cette adaptation :

- exige davantage que la seule continuité de service (propriété que le Génie Logiciel étudie pourtant depuis une décennie) : l'utilité, l'utilisabilité et, par extension, la valeur doivent être garanties. Or ces requis de qualité, du point de vue de l'utilisateur, sont fonction du ... contexte d'usage. Autrement dit, ils ne peuvent être fixés a priori, mais doivent être modélisés explicitement en sorte que le système puisse les évaluer au cours du processus d'adaptation ;

- doit être contrôlable, ni trop, ni trop peu, par l'utilisateur (alors que la recherche en systèmes répartis et robotique vise à exclure l'utilisateur de la boucle !). Il convient donc d'étudier avec attention les fonctions de la méta-IHM et son insertion dans les processus d'adaptation ;

- se traduit par un remodelage et/ou une redistribution de tout ou partie de l'IHM du système, avec un état de reprise allant de la session à l'action physique ; IHM, adaptable ou adaptée, dont le déploiement peut être statique ou dynamique, dont la mise en œuvre peut faire intervenir ou non plusieurs espaces technologiques. Concernant le remodelage, celui-ci peut aller du cosmétique de surface (c'est-à-dire limité à la Présentation Physique) à des remaniements profonds du noyau

<sup>9</sup> Observabilité : capacité du système à rendre perceptibles, à tout instant, les concepts et fonctions métiers nécessaires et suffisants pour que l'utilisateur puisse évaluer l'état actuel du système et/ou pour qu'il puisse accomplir les tâches possibles sans erreur [IFI 96].

<sup>10</sup> Inter-utilisabilité : « la facilité avec laquelle les utilisateurs transfèrent et adaptent ce qu'ils ont appris lors de leurs précédentes utilisations d'un service lorsqu'ils y accèdent avec un nouveau support » [KAR 05].

fonctionnel. Le remodelage peut jouer aussi sur la multimodalité, aspect très peu exploité jusqu'ici en plasticité des IHM [PAT 08] [SER 09] [DUA 06].

Nous venons de présenter en détail les dimensions de la plasticité des IHM et ses requis de qualité. Dans les sections qui suivent, nous considérons sa mise en œuvre technique en deux volets : cadre(s) de référence CAMELEON (9.3 et 9.4) et nos recommandations pour la mise en œuvre (9.5).

### 9.3. Le cadre de référence CAMELEON pour un développement rationnel des IHM plastiques

Le cadre de référence CAMELEON (Figure 9.13) explicite et formalise les différents aspects qui concourent à la production d'IHM plastiques en termes de modèles et de relations entre modèles<sup>11</sup>. Il est en cela conforme aux principes de l'Ingénierie Dirigée par les Modèles (IDM) du Génie Logiciel. Nous reprenons en détail ces deux points – modèles et relations en 9.3.1 et 9.3.2 respectivement, puis nous montrons en 9.3.3 comment les exploiter pour un développement rationnel.

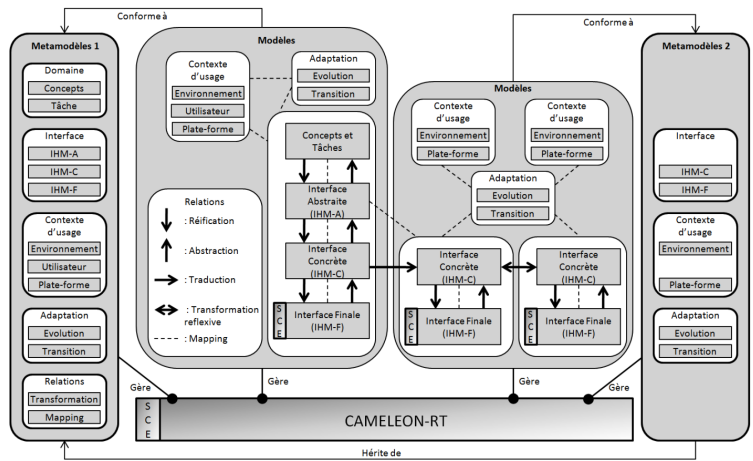
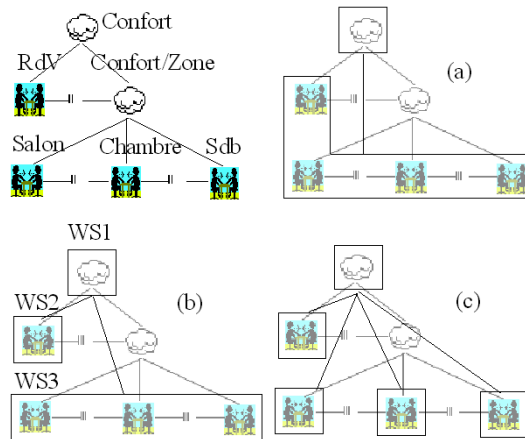


Figure 9.13 : Le cadre de référence CAMELEON (adapté de [CAL 03]).

<sup>11</sup> Ce cadre de référence tient son nom du projet européen IST-2000-30104 CAMELEON (Context Aware Modelling for Enabling and Leveraging Effective interaction) – 2001-2004, <http://giove.isti.cnr.it/projects/cameleon.html>, au cours duquel il a été conçu. En vérité, [CAL 01] est la première publication de ce cadre de référence qui, alors, ne portait pas de nom.

Rappelons qu'un modèle d'entité est une représentation de cette entité conçue pour un objectif donné. Dans notre cas, les entités sont des IHM (ou des éléments d'IHM) et l'objectif est la production opérationnelle d'IHM plastique. Comme le montre la Figure 9.13, les modèles recommandés par le cadre de référence CAMELEON sont les suivants : modèle de tâche et modèle des concepts métier, IHM abstraite, IHM concrète, IHM finale, modèle du contexte d'usage et modèle d'adaptation.

- Le modèle de tâche (MT) et le modèle des concepts métier décrivent respectivement les procédures pour atteindre les buts et sous-but que l'utilisateur se fixe avec le système et les entités dépendantes de l'application manipulées au cours de l'accomplissement de ces tâches. La Figure 9.14 en haut à gauche montre un exemple de modèle de tâche correspondant à la gestion du confort de son domicile.



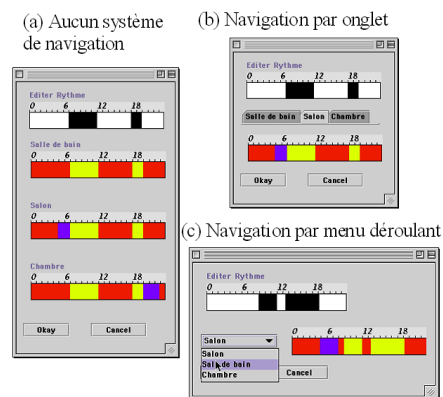
**Figure 9.14** : Trois IHM-A possibles pour un même arbre de tâches : « Programmer son confort » (exemple extrait de [THE 01]). Cette tâche consiste à définir son rythme de vie (RdV) et programmer son confort par zone. Trois zones (c'est-à-dire trois concepts métiers) sont identifiées : le salon, la chambre et la salle de bain. En (a), les quatre tâches feuilles sont hébergées dans un même espace de travail élémentaire (WS pour WorkSpace). En (b), éditer rythme de vie est isolé dans un espace élémentaire dédié (WS2) tandis que les trois autres tâches sont regroupées dans WS3. En (c), un espace de travail élémentaire est associé à chaque tâche feuille du modèle de tâche. Dans les trois cas, les espaces de travail élémentaires sont accessibles de l'espace de travail correspondant à la tâche racine (WS1).

- L'IHM abstraite (IHM-A) décrit la structuration de l'IHM en termes d'espaces de travail (appelés aussi espaces de dialogue) et de navigation entre ces espaces,

aussi indépendamment que possible de la plate-forme cible. Autrement dit, à ce niveau de représentation d'une IHM, le « look and feel » de l'IHM est indéterminé. La Figure 9.14 montre une représentation graphique de plusieurs IHM-A possibles (a, b, c) correspondant au modèle de tâche de la gestion du confort.

- L'IHM concrète (IHM-C) représente l'IHM en termes d'interacteurs (par exemple, des boutons, menus, images, etc.). Ici, le « look and feel » est fixé. La Figure 9.15 montre des IHM-C correspondant aux IHM-A et au modèle de tâche de la Figure 9.14.

- L'IHM finale (IHM-F) désigne l'IHM telle que l'utilisateur la perçoit et la manipule à l'exécution. Bien que l'IHM-C et l'IHM-F soient conceptuellement proches, l'expérience montre que le même code source (Java ou HTML par exemple) n'a pas le même comportement à l'exécution selon les JVM et les navigateurs.



**Figure 9.15 :** Exemples d'IHM-C produites à partir des IHM-A de la figure 9.14 (exemple extrait de [THE 01]). (a) IHM-C produite à partir de l'IHM-A (a) où toutes les tâches feuilles sont transformées en un seul espace de travail. Cette IHM-C aurait pu être produite aussi à partir de l'IHM-A (c) dans laquelle tous les espaces de travail élémentaires sont simultanément visibles dans l'espace de travail parent. (b) et (c) correspondent respectivement aux IHM-C sur téléphone portable et PDA. Seuls changent les interacteurs de navigation entre les espaces de travail. Ces deux IHM-C peuvent être produites à partir des IHM-A (b) et (c) de la Figure 9.14 où l'espace de travail correspondant à la gestion du rythme de vie est toujours visible.

- Le *contexte d'usage* inclut le modèle de l'utilisateur, le modèle de la plate-forme et le modèle de l'environnement physique et social (cf. section 9.2.3).

- Le modèle d'adaptation comprend le modèle d'évolution qui spécifie la nature de l'adaptation en réponse au changement de contexte et le modèle de transition qui exprime la nature de l'IHM de transition qui accompagne l'adaptation en sorte que l'utilisateur puisse évaluer et contrôler l'état de ce processus. Notons que ces deux modèles – modèle d'évolution et IHM de transition, sont observables, contrôlables, voire programmables par une méta-IHM (cf. section 9.2.4).

### 9.3.2. Les relations entre modèles

Tous les modèles du cadre de référence CAMELEON entretiennent trois classes de relations que la figure 9.13 met en évidence : conformité, transformation, mapping.

Relation de conformité. Tout modèle est censé être conforme à un métamodèle. Un métamodèle décrit un langage qui permet de produire des (instances de) modèles. Par exemple, le modèle de tâche de la figure 9.14 est conforme au métamodèle CTT [PAT 03a]. L'ontologie présentée pour le contexte en section 9.2.3.1 est un métamodèle de contexte.

```

module M2TaskToM2Workspace {
  from M1Task : M2Task
  to M1Workspace : M2Workspace
  -- One workspace per task
  rule TaskToSpace {
    from t : M2Task!Task
    to w : M2Workspace!Space (
      name <- t.name )
  }

  -- OrOperator to SequenceOperators
  rule OrOperatorToSequence{
    from o : M2Task!BinaryOperator (
      o.name = "or"
    )
    to motherToLeft : M2Workspace!Sequence (
      origin<- [TaskToSpace.w]o.motherTask,
      destination<-[TaskToSpace.w]o.leftTask
    )
  }
}

```

**Figure 9.16 :** Exemple de transformation ATL d'un modèle de tâche M1Task (conforme au métamodèle M2Task) vers l'IHM-A M1Workspace (conforme au métamodèle M2Workspace (exemple extrait de [SOT 08]). La règle TaskToSpace crée un espace de travail w par tâche source t où w prend le nom de t. w et t sont des éléments des modèles M1Task et M1Workspace. La règle OrOperatorToSequence transforme tout opérateur de type OR o entre deux tâches (o.leftTask et o.rightTask) en opérateur de séquence entre l'espace de travail parent et ses filles (de o.motherTask vers o.leftTask et de o.motherTask vers o.rightTask).

Relation de transformation. Une transformation est la production d'un ensemble de modèles cibles à partir d'un ensemble de modèles sources. Cette transformation est définie par un ensemble de règles qui spécifient comment l'ensemble source est transformé en l'ensemble cible. Notons que cet ensemble de règles constitue à son tour un modèle conforme à un métamodèle. QVT (Query/View/Transformation), un standard de l'OMG, et ATL (A Transformation Language) [BEZ 03], développé au-dessus d'Eclipse, sont deux exemples de métamodèles d'expression de transformations de modèles. La figure 9.16 montre un exemple de règles ATL qui, à toute tâche du modèle de tâche fait correspondre un espace de travail ainsi que les enchaînements entre ces espaces de travail. La figure 9.16 illustre sous forme de schémas, le principe des transformations des opérateurs du MT sous forme d'espaces de travail avec leur enchaînement.

On distingue :

- les transformations de réification qui réduisent le niveau d'abstraction des modèles (par exemple, du modèle de tâche à l'IHM-C). Typiquement une IHM-A est obtenue par une transformation de réification du MT et des concepts métier (en lien éventuellement avec d'autres modèles comme le modèle du contexte). Les IHM-A de la figure 9.14 résultent de la transformation du modèle de tâche de gestion du confort en appliquant différentes règles de transformation.

- les transformations d'abstraction qui élèvent le niveau d'abstraction (par exemple, d'une IHM-C vers une IHM-A).

- les transformations de traduction qui maintiennent le niveau d'abstraction des modèles source (par exemple, passer d'une IHM-A pour un contexte donné vers une IHM-A pour un autre contexte).

- les transformations réflexives qui remplacent un modèle donné par une nouvelle instanciation de lui-même [BOU 02].

- les *relations de mapping* qui permettent de tracer la correspondance entre éléments de modèles sources et cibles (et vice versa). Alors que la relation de transformation exprime une dépendance globale entre modèles sources et cibles, le mapping correspond à un besoin de traçabilité entre éléments de modèles. Par exemple, toute tâche T d'un modèle de tâche et l'ensemble des concepts métiers C qu'elle référence sont transformés en un espace de travail WS qui inclut une représentation abstraite des concepts de C. La correspondance entre T et WS est représentée par une fonction de mapping. La correspondance entre les concepts de C et leur représentation abstraite dans WS est maintenue par une fonction de mapping. Et ainsi de suite entre WS et sa transformation en un interacteur fenêtre ou canevas dans l'IHM-C et entre les représentations abstraites des concepts métiers de WS et les interacteurs nécessaires à leur représentation concrète dans l'IHM-C.

En pratique, comment agencer ces modèles et relations dans un processus de développement ?

### 9.3.3. Processus de développement

En Interaction Homme-Machine, il est recommandé de produire l'IHM finale d'un système interactif par réifications successives avec, comme point de départ, les modèles de tâche et des concepts métier. Cette approche fait l'hypothèse forte que les concepteurs-développeurs savent produire un modèle de tâche, ce qui, en pratique, est loin d'être le cas. Par ailleurs, la demande actuelle est largement centrée sur la rétro-conception d'IHM-finales en sorte qu'elles soient disponibles à la fois sur station de travail et sur SmartPhones. Autrement dit, il n'est pas de processus de production unique mais une approche rationnelle adaptée au cas par cas.

Le cadre de référence CAMELEON répond à cette exigence de souplesse avec le concept de point d'entrée et de composition idoine de transformations. Par exemple, une IHM-C conçue pour un contexte d'usage donné est transformée par abstraction, éventuellement en une seule étape, en un modèle de tâches qui correspond à ce contexte. Ce modèle de tâches est ensuite traduit en un modèle de tâches adapté à un autre contexte cible, puis une nouvelle IHM-C est produite par réifications successives. La figure 9.17 illustre un sous-ensemble des possibilités.

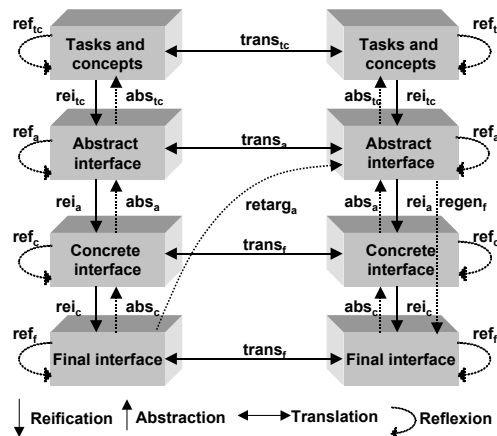
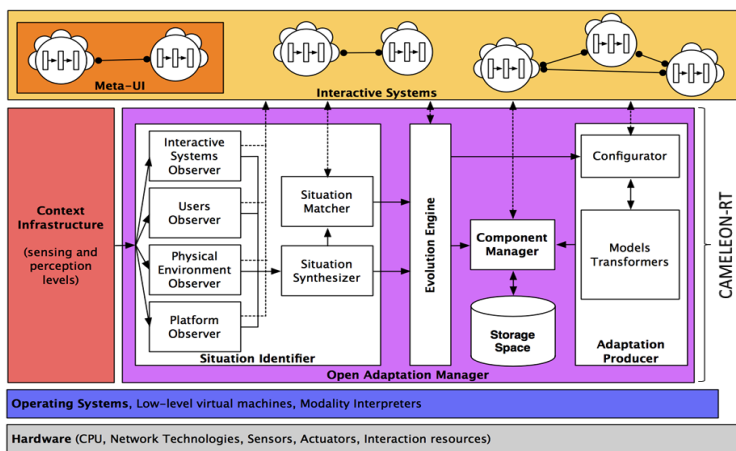


Figure 9.17 : Processus de conception par composition de transformations (d'après [LIM 04]).

Tout ou partie des modèles que nous venons de passer en revue sont produits en phase de conception du système interactif pour nourrir in fine la phase d'exécution. Celle-ci, par définition, nécessite une infrastructure d'exécution distincte techniquement des systèmes interactifs.

#### 9.4. Infrastructure d'exécution CAMELEON-RT

Le cadre de référence CAMELEON original dit peu de choses sur l'infrastructure d'exécution si ce n'est qu'elle inclut des mécanismes « SCE » répartis entre l'IHM-Finale du système interactif et l'infrastructure d'exécution.



**Figure 9.18 :** CAMELEON-RT. Décomposition fonctionnelle canonique de l'infrastructure d'exécution pour IHM plastiques (d'après [BAL 04]). Dans ce modèle en couches, les côtés supérieurs des boîtes offrent une API à destination des couches supérieures. Les flèches continues dénotent les échanges directionnels d'information. Les flèches en pointillé indiquent les fonctions rendues accessibles par API aux couches supérieures. Les symboles en forme de fleur représentent des composants logiciels. Les segments terminés par de petits disques noirs symbolisent des liaisons entre composants.

Un mécanisme « SCE » se décompose ainsi :



- S (pour Sensing) dénote les fonctions de reconnaissance de la situation et de détection d'un changement de situation ;
- C (pour Compute) fait référence au calcul de la réaction au changement de situation, c'est-à-dire au calcul de l'adaptation à mettre en œuvre si besoin ;
- E (pour Execute) se réfère à l'exécution de la réaction.

La nature transversale de l'adaptation explique que les mécanismes SCE soient répartis entre l'IHM finale et l'infrastructure d'exécution. C'est l'effet *slinky*<sup>12</sup> introduit en architecture logicielle des systèmes interactifs [ARC 92].

Cette section décrit CAMELEON-RT (pour CAMELEON Run-Time) [BAL 04], l'infrastructure d'exécution que nous avons proposée comme complément du cadre CAMELEON original.

#### **9.4.1. Décomposition fonctionnelle de CAMELEON-RT**

La Figure 9.18 montre la décomposition fonctionnelle de CAMELEON-RT. Au-dessus de CAMELEON-RT, s'exécutent des systèmes interactifs à IHM plastiques. En dessous, se trouvent les couches système d'exploitation et matérielles natives (logiciels de base).

La couche « matériel » représente l'ensemble des plates-formes élémentaires qui composent la plate-forme de l'utilisateur (son « cloud » personnel). Ces plates-formes élémentaires se caractérisent par la puissance et le type de leur microprocesseur, la bande passante offerte par leur technologie réseaux, les capteurs et actuateurs qu'elles contrôlent et les ressources d'interaction dont elles disposent. Pour rappel, ces caractéristiques alimentent, sous forme d'observables, le modèle de la plate-forme dans le contexte d'usage.

Sur la couche « matériel » repose la couche « système d'exploitation ». Ici, la notion de système d'exploitation est à prendre au sens large : machines virtuelles, par exemple JVM ou .Net, interpréteurs de modalités ou boîtes à outils spécialisées

---

<sup>12</sup> Par analogie au fonctionnement du jouet Slinky inventé en 1943, ce terme a été introduit avec le modèle d'architecture Arch pour exprimer le transfert (l'allocation) de fonctions entre composants selon les critères architecturaux retenus [ARC 92]. En effet, une fois mis en mouvement, ce jouet, qui se présente sous la forme d'un ressort, voit sa masse se déplacer dynamiquement. Voir <http://www.poof-slinky.com/Slinky-Museum/Slinky-History/>

dans le rendu d'interfaces utilisateur (par exemple, Swing<sup>13</sup>, OpenGL<sup>14</sup>, DirectSound<sup>15</sup> et ALSA<sup>16</sup>).

Au-dessus des couches « matériel » et « système d'exploitation » se trouve CAMELEON-RT, la couche intergiciel dédiée à la plasticité. Elle comprend :

- une infrastructure de capture du contexte d'usage qui couvre les niveaux capture et transformation du contexte [REY 05] [COU 05]. Cette infrastructure alimente le gestionnaire de l'adaptation ;
- le gestionnaire de l'adaptation qui assure les trois fonctions SCE : l'identificateur de situation alimente un moteur d'évolution en charge de composer un plan d'adaptation avec l'aide du gestionnaire de composants. Ce plan est transmis au producteur de l'adaptation qui le met en œuvre.

Nous détaillons plus avant le gestionnaire de l'adaptation dans les sous-sections qui suivent.

#### **9.4.2. Identificateur de situation**

L'identificateur de situation comprend deux classes de fonctions : d'une part, les observateurs et, d'autre part, les synthétiseur et comparateur de situation.

Les observateurs sont chargés de filtrer et de combiner les observables symboliques issus de l'infrastructure de capture du contexte pour alimenter le synthétiseur de situation au bon niveau d'abstraction. Les observateurs se répartissent en quatre classes : les observateurs des systèmes interactifs et de la méta-IHM en cours d'exécution (par exemple, quelle est la tâche courante ? Quels en sont les composants ?), les observateurs de la plate-forme, les observateurs de l'utilisateur et les observateurs de l'environnement physique et social couvrant ainsi l'ensemble des observables symboliques caractéristiques des trois dimensions du contexte d'usage (cf. Section 9.2.3).

Les observables produits par les quatre classes d'observateurs sont interprétés sous forme d'une situation par le synthétiseur de situation. Cette situation modélise l'état courant du contexte dans lequel se déroule l'interaction. Il revient au synthétiseur de situation de détecter le changement d'une situation vers une autre et d'un contexte vers un autre (Cf. Section 9.2.3). La modélisation de la nouvelle situation courante est ensuite transmise au moteur d'évolution d'une part et au

---

<sup>13</sup> <http://java.sun.com/javase/6/docs/technotes/guides/swing/index.html>

<sup>14</sup> <http://www.opengl.org>

<sup>15</sup> <http://www.microsoft.com>

<sup>16</sup> <http://www.alsa-project.org>

comparateur de situation d'autre part. Le comparateur de situation vise à établir une correspondance avec une situation d'interaction connue (prévue à la conception ou rencontrée par le passé).

#### **9.4.3. Moteur d'évolution**

Si la nouvelle situation est connue, le moteur d'évolution est en mesure d'appliquer, soit les règles d'adaptation définies par le concepteur ou par l'utilisateur par le biais d'une méta-IHM, soit celles qui auront été inférées lors des expériences précédentes. Si le comparateur de situation est incapable d'établir une correspondance avec une situation d'interaction connue, le moteur d'évolution met en œuvre d'autres types de mécanismes pour proposer un plan d'adaptation du système interactif, voire faire appel à l'utilisateur par le biais de la méta-IHM. Dans tous les cas, le moteur d'évolution fait appel au gestionnaire de composants pour élaborer un plan d'adaptation.

#### **9.4.4. Gestionnaire de composants**

Le gestionnaire de composants fait l'interface entre un espace de stockage de composants et ses clients (c'est-à-dire le moteur d'évolution, le producteur de l'adaptation et les systèmes interactifs). Sa fonction correspond à celle de l'annuaire de services utilisé dans les approches à services : il permet l'enregistrement et la découverte dynamique de composants. Afin que les clients puissent formuler des requêtes de recherche de composants, ceux-ci sont métadécrits (ce qui signifie qu'il existe un métamodèle de composant). Les clients du gestionnaire de composants formulent des requêtes qui font référence aux propriétés fonctionnelles et extra-fonctionnelles du type de composants recherchés. Par exemple, le moteur d'évolution recherche un composant capable de permettre telle tâche utilisateur (propriété fonctionnelle), capable de s'insérer dans un système interactif conçu selon tels architecture logicielle et environnement d'exécution (propriété extra-fonctionnelle) et utilisant telle modalité d'interaction (propriété extra-fonctionnelle).

#### **9.4.5. Producteur de l'adaptation**

Le plan d'adaptation fourni par le moteur d'évolution fait référence aux composants retenus ainsi que la description des liaisons qu'il convient d'établir ou de détruire. Suivant l'état des composants référencés (inertes ou instanciés) et suivant leur nature (exécutable ou transformable), le producteur de l'adaptation met en œuvre les mécanismes idoines de transformation, d'instanciation ou de suppression de composants, puis de suppression et d'établissement de liaisons.

Un composant inerte réside dans la base de composants. Il n'est pas encore installé (instancié) en mémoire. Un composant exécutable existe sous forme de code (compilé ou interprétable). Un composant transformable (par exemple un composant qui encapsule un modèle de tâche) doit être transformé pour devenir exécutable. Ainsi un composant inerte exécutable demande une étape de chargement pour devenir instancié exécutable. Un composant inerte transformable doit être chargé (il devient instancié transformable), puis une fois transformé passe dans l'état instancié exécutable. Notons que la nature du plan dépend de la nature logicielle du système interactif à adapter.

La nature logicielle des systèmes interactifs se répartit en deux classes : la classe « traditionnelle » comprend les systèmes interactifs développés suivant un processus de conception classique où les modèles de conception sont progressivement réifiés pour obtenir un programme exécutable. La deuxième classe, orientée IDM, regroupe les systèmes interactifs développés suivant une approche dirigée par les modèles où les modèles de haut niveau d'abstraction ne sont pas « consommés » par le processus de réification, mais, au contraire, sont sémantiquement liés les uns aux autres puis embarqués au sein du programme exécutable final en vue d'être utilisés pendant son exécution. Afin de prendre en considération les moyens d'adaptation de ces deux classes de systèmes interactifs, la fonction producteur de l'adaptation se décompose en deux sous-fonctions : le transformateur de modèles, consacré à la transformation de modèles pour l'IHM et le configurateur destiné à l'adaptation par le biais d'API (Application Programming Interface). Soit, il s'agit d'API offerte par le système interactif et le configurateur doit être conçu selon cette API. Il s'agit alors d'une solution ad hoc. Soit l'API est offerte par l'infrastructure et les systèmes interactifs sont conçus en tenant compte de cette API. Il s'agit alors d'une solution normalisée générique.

En synthèse, le cadre CAMELEON original et son complément CAMELEON-RT constituent un modèle canonique qui sert à la fois la conception et l'exécution de la famille des IHM plastiques. De nombreux outils de mise en œuvre s'y réfèrent et l'appliquent de diverses manières. Une synthèse est proposée en annexe 2.

### **9.5. Nos principes de mise en œuvre de la plasticité**

Nous préconisons l'application des principes suivants pour le développement technique de systèmes interactifs à IHM plastiques ou d'outils permettant la réalisation de tels systèmes.

1) Choisir un espace problème (comme celui de la Figure 9.2) de façon à formuler avec précision les requis de l'espace solution. De même, choisir un système de référence pour mesurer la qualité ergonomique du système cible (par exemple, le cadre Bastien-Scapin [BAS 93]).

2) Utiliser le cadre de référence CAMELON et CAMELEON-RT comme espace solution de base : modèles conformes à des métamodèles ainsi que la décomposition fonctionnelle de l'infrastructure d'exécution sont les unités canoniques auxquelles il convient de se référer.

3) Retenir du cadre de référence CAMELEON et CAMELEON-RT seulement les unités canoniques nécessaires et suffisantes à la satisfaction des requis de l'espace solution. Par exemple, il est possible d'éliminer le modèle de tâches si l'on s'en tient à du remodelage de surface (niveau Présentation Physique). Bien noter que : (1) les transformations et la qualité doivent être des modèles, ce qui permet de contrôler finement les transformations au regard de la qualité ergonomique visée ; (2) les composants d'IHM enregistrés dans la base de composants, s'il doit y en avoir, doivent être métadécrits conformément à un métamodèle de composants [BOU 07], ce qui permet de les recruter à la demande. Ces composants ne sont pas toujours directement exécutables (section 9.4). Il peut s'agir de composants transformables, tels un modèle de tâches ou une IHM-A.

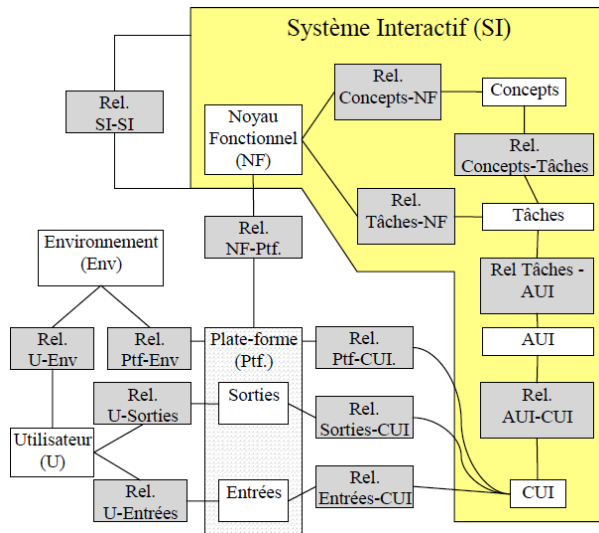


Figure 9.19 : Un système interactif est un graphe de modèles.

4) Estomper la distinction entre phases de conception et d'exécution<sup>17</sup>. Plus précisément : tout modèle productif spécifié par un concepteur, de même, les modèles résultant de transformations ainsi que les mappings entre modèles et éléments de modèles, doivent être accessibles en phase d'exécution. En corollaire, comme le montre la Figure 9.19, un système interactif exécutable n'est pas seulement du code mais un graphe de modèles dans lequel chaque modèle décrit un aspect particulier pertinent du système. Typiquement, l'existence des modèles de tâches et des concepts métiers permettent de réaliser des adaptations dynamiques fondées sur des connaissances sémantiques [BLU 10]. Sans eux, l'adaptation se limite de facto à des modifications de surface (niveaux PP et PL).

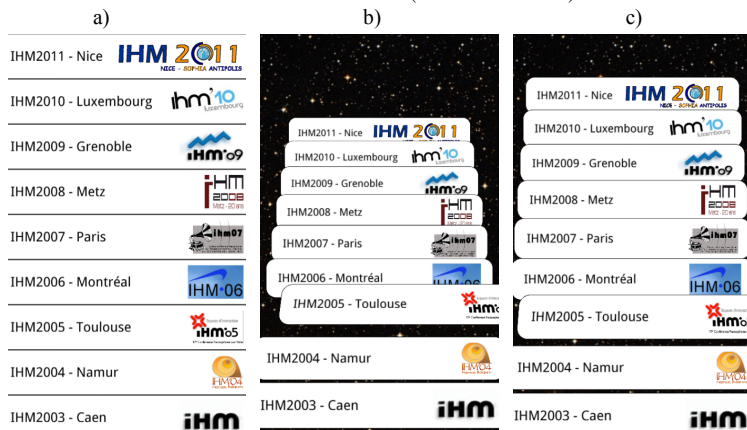


Figure 9.20 : Transition (b) d'un menu 2D (a) vers un menu 3D (c).

5) Décider de la répartition (effet slinky) des fonctions SCE entre le système interactif et l'infrastructure d'exécution. (Pour rappel : S=identification et détection de changement de situation, C=Calcul du plan d'adaptation et E=Exécution du plan). Cette décision est largement liée à la couverture SCE des outils d'implémentation. Par exemple, dans le cas d'une implémentation du système au moyen de COMETS [DEM 07] [DEM 08], les fonctions SCE sont embarquées en totalité dans le système interactif. À l'autre extrême, dans le cas de PhotoBrowser, les fonctions SCE sont entièrement assurées par l'infrastructure d'exécution Ethylène [BAL 08].

<sup>17</sup> En Génie Logiciel, la dichotomie entre les phases de conception et d'exécution tend à s'estomper [FER 09]. Nous recommandons une voie similaire pour les IHM plastiques.

6) Ne pas négliger les IHM de transition. Là encore, nous sommes tributaires des outils de mise en œuvre. Par exemple, la boîte à outils NOMAD offre la possibilité de spécifier un comportement de morphing au niveau de widgets (Figure 9.20).

### 9.6. Conclusion : les acquis et verrous

La plasticité des IHM est un problème multidimensionnel complexe que l'on essaie de résoudre depuis une bonne dizaine d'années. Certains acquis sont solides, mais tous les verrous n'ont pas sauté. Parmi les acquis, il faut compter les définitions, l'espace problème et le cadre de référence CAMELON. Si les éléments conceptuels sont plutôt bien partagés au sein de la communauté IHM, les solutions techniques sont diverses et parcellaires. Toutefois, l'approche IDM, qui apporte avec elle l'automatisation et la rigueur, a de fervents adeptes. Les avantages de l'IDM sont directement observables tant en phase de conception (allègement de la tâche du développeur et donc diminution des coûts de développement) qu'en phase d'exécution (adaptation autonome dynamique). Toutefois, en l'état du savoir-faire, l'approche tout-IDM n'a pas tout résolu.

Nous voyons plusieurs limites au tout-IDM :

1) L'IDM, en tant que méthodologie de développement, a favorisé la dichotomie entre phases de conception et d'exécution. Or, le cadre de référence CAMELEON montre qu'un système interactif est un graphe de modèles et donc qu'un système interactif n'est pas seulement du code exécutable. En corollaire, phases de conception et d'exécution sont un même combat !

2) Absence de norme concernant les métamodèles même si UML [USI 11] [GAR 11] vise cet objectif. La profusion d'initiatives sur le développement de langages de description d'interfaces utilisateur est symptomatique de la difficulté à définir un ensemble de métamodèles qui soit cohérent, non-ambigu, facile à comprendre et complet (c'est-à-dire capable de couvrir l'espace problème des IHM plastiques). Autrement dit, nous avons les bons modèles (« the right models ») mais pas encore les modèles bons (« the models right »).

3) De notre point de vue, a minima trois métamodèles sont à la clef de la réussite de l'IDM en plasticité des IHM : les métamodèles de transformation, de qualité et d'IHM-C.

Les transformations et leur composition constituent un outil théorique élégant et générique ouvrant la voie à toute forme de processus de développement et d'intégration technique. Toutefois, les transformations sont difficiles d'accès : les langages (métamodèles) comme ATL ou QVT ne sont accessibles qu'à des programmeurs confirmés et motivés ! Mais aussi, ce qui est essentiel pour nous, communauté IHM, ces langages ne permettent pas d'exprimer directement des

règles d'utilisabilité [SOT 07] [GAR 11]. Plus fondamentale est l'impossibilité de déduire automatiquement, à partir de la spécification d'une transformation, sa transformation inverse. On assiste là à une faille majeure qui peut conduire à des modèles incohérents par suite de réifications et d'abstractions appliquées de manière itérative au cours du cycle de vie d'un système interactif.

Au niveau IHM-C, les métamodèles actuels, non seulement sont en retard sur l'innovation, mais brident la créativité. Les langages pour l'expression d'IHM-C suivent la technologie au lieu d'ouvrir de nouvelles pistes. Alors que les propriétés CARE [COU 95] ont été introduites il y a plus de quinze ans et que le célèbre paradigme « mets ça là » [BOL 80] a plus de 30 ans, les langages de description d'IHM-C peinent encore à décrire les IHM multimodales. Certes, nous sommes en mesure de générer des IHM multimodales simplistes (par exemple, avec XHTML+VoiceXML), mais sans véritable microdialogue de réparation. En vérité, nous en sommes encore à définir le langage qui décrira de manière convenable les IHM graphiques à manipulation directe. Pendant ce temps :

- de nouvelles formes d'ordinateurs à construire soi-même, tels les siftables du MIT et les toy blocks de CMU, arrivent sur le marché ;
- de nouvelles formes d'interaction prolifèrent de toute part que ce soit pour répondre aux situations de mobilité (par exemple, SixthSense [MAE 09] [MIS 09]), pour interagir en 3D (où gestes 3D et écrans 3D commencent à s'imposer), pour l'interaction sur tables interactives ou encore pour l'interaction multisurface [BAL 09] ;
- de nouveaux requis de qualité se font jour : le design passe du développement de systèmes utiles et utilisables pour des utilisateurs à buts précis dans un contexte précis à la conception d'espaces de création où l'utilisateur passe du statut de consommateur au statut d'inventeur.

En synthèse, il convient que les métamodèles de niveau IHM-C capturent l'expression de la convergence sans limite entre le numérique et le physique. Peut-être, la métamodélisation est par essence, la mauvaise approche pour ce qui concerne les IHM-C : un modèle, qui représente une chose, est nécessairement une simplification, donc une réduction de cette chose réelle. Dès lors, les aspects subtils de l'interaction, qui font toute la différence entre un design inspiré et un design ordinaire, sont probablement mieux exprimés en utilisant le code exécutable directement plutôt qu'une abstraction de ce code. Dès lors, ce code doit se présenter sous forme de service de façon à en permettre le recrutement dynamique. Mais aussi, il faut tenir compte de ce constat : les designers excellent à traduire en dessin des solutions d'interaction, mais ils ont quelques difficultés à exprimer le comportement dynamique de ces dessins, les conduisant à s'exprimer en langage naturel [MYE 08]. Une piste prometteuse pour réconcilier la pratique des designers et l'intérêt du code est de remettre au goût du jour les travaux à la Peridot [MYE 90]



qui, comme SketchiXML [COY 04] [KIE 10] font de la rétro-ingénierie de dessins vers des IHM concrètes calculables. Toutefois, l'inférence de comportement à partir d'exemples, à la façon du prometteur « Watch What I Do » lancé à la fin des années soixante-dix (cf. le système Pygmalion de Dave Smith's [SMI 93]), reste une question ouverte.

Un dernier problème non traité explicitement dans ce chapitre est la (re)composition dynamique du Noyau Fonctionnel qui implique la (re)composition de son IHM. Cet aspect est encore à peine esquissé pour les technologies du Web [CAN 09] [PIE 09] [YU 07]. Il fait l'objet du chapitre XX.

## Bibliographie

- [ABO 92] ABOUW D.G.D., COUTAZ J., NIGAY L., Structuring the Space of Interactive System Properties », dans Larson J. et Unger C. (dir), *Engineering for Human-Computer Interaction*, Elsevier Science Publishers, Hollande, 1992, pp 113-126.
- [ALL 06] ALLEN C., WALLACH W., SMITH I., *Why Machine Ethics?*, IEEE Intelligent Systems, 21(4), 2006, pp. 12-17.
- [ARC 92] Arch, *A Metamodel for the Runtime Architecture of An Interactive System*, *The UIMS Developers Workshop*, SIGCHI Bulletin, 24(1), ACM, January, 1992.
- [BAI 08] BAILLY G., DEMEURE A., NIGAY L., LECOLINET E., *MultiTouch Menu*, IHM 2008, 20ème conférence francophone sur l'Interaction Homme-Machine, ACM Press, 2008.
- [BAL 04] BALME L., DEMEURE A., BARRALON N., COUTAZ J., CALVARY G., *CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces*, Lecture Notes in Computer Science, Vol. 3295, 2004. In P. Markopoulos, B. Eggen, E. Aarts et al. (Eds), *Ambient intelligence: Second European Symposium, EUSAI 2004*. Springer-Verlag Heidelberg (Publisher), ISBN: 3-540-23721-6, Eindhoven, The Netherlands, November 8-11, 2004, pp 291-302.

- [BAL 08] BALME L., *Interfaces Homme-Machine Plastiques : Une approche par composants dynamiques*, Thèse de l'Université Joseph Fourier, Grenoble I, 2008.
- [BAL 09] BALAKRISHNAN R., BAUDISCH P., *Special Issue on Ubiquitous Multi-Display Environments, Human-Computer Interaction*, Vol. 24, Taylor and Francis, 2009.
- [BAR 97] BARDRAM J.E., *Plans as Situated Action: An Activity Theory Approach to Workflow Systems*, Proceedings of ECSCW'97 Conference, Lancaster UK, September 1997.
- [BAS 93] BASTIEN J.M.C., SCAPIN D., *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*, Rapport technique INRIA, n°156, June 1993.
- [BAU 10] BAU O., *Interaction streams: helping users learn, execute and remember expressive interaction grammars*. thèse doctorat, Université Paris-Sud, 2010.
- [BER 99] BERARD F., *Vision par ordinateur pour l'interaction homme-machine fortement couplée*, Thèse de Doctorat de l'Université Joseph Fourier, Grenoble, France, 2009.
- [BEY 98] BEYER H., HOLTZBLATT K., *Contextual Design: Defining Customer-centered Systems*, Morgan Kaufman, 1998.
- [BEZ 03] BEZIVIN J., DUPE G., JOUAULT F., PITETTE G., ROUGUI J., *First experiments with the ATL transformation language: transforming XSLT into Xquery*. OOPSLA Workshop, 2003.
- [BLA 01] BLACKWELL A.F., HAGUE R., *AutoHAN: An architecture for programming the home*, In Proc. Of the IEEE Symposium on Human-Centric Computing Languages and Environments, Stresa, Italie, 2001, pp 150-157.
- [BLA 11] BLANCH R., ORTEGA M., *Benchmarking Pointing Techniques with Distractors: Adding a Density Factor to Fitts' Pointing Paradigm*, In CHI 2011: Proceeding of the twenty-ninth annual SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM, 2011.
- [BLU 10] BLUMENDORF M., LEHMANN G., ALBAYRAK S., *Bridging models and systems at runtime to build adaptive user interfaces*. In Proc. of the 2010 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2010, ACM Publ., 2010, pp 9-18.
- [BOL 80] BOLT R., *"Put That There": Voice and gesture at the graphics interface*. In Proc. of the 7th International Conf. on Computer Graphics and Interactive techniques, ACM Publ., 1980, pp 262-270.
- [BOU 02] BOUILLON L., VANDERDONCKT, J., *Retargeting web pages to other computing platforms*. Proceedings of IEEE 9th Working Conference on Reverse Engineering WCRE'2002 (Richmond, 29 October-1 November 2002), IEEE Computer Society Press, Los Alamitos, 2002, pp 339-348.
- [BOU 07] BOURGUIN G., LEWANDOWSKI A., TARBY J.-C., *Defining Task Oriented Component*. In Proc. TAMODIA 2007, Lecture Notes in Computer Science 4849 Springer 2007, ISBN 978-3-540-77221-7, 2007, pp 170-183.
- [BRA 03] BRANGIER E., BARCENILLA J., *Concevoir un produit facile à utiliser*, Éditions d'Organisation, Paris, 2003.
- [BRE 02] BREZILLON P., *Expliciter le contexte dans les objets communicants*. Les Objets Communicants. Hermes Sciences Editions, Lavoisier, Chapitre 21, 2002, pp 295-303.

- [CAL 01] CALVARY G., COUTAZ J., THEVENIN T., *A Unifying Reference Framework for the Development of Plastic User Interfaces*, in IFIP WG2.7 (13.2) Working Conference, Engineering Human Computer Interaction (EHCI01), Toronto, Springer Verlag Publ., LNCS 2254, 2001, pp 173-192.
- [CAL 02] CALVARY G., COUTAZ J., BOUILLON L., FLORINS M., LIMBOURG Q., MARUCCI L., PATERNÒ F., SANTORO C., SOUCHON N., THEVENIN D., VANDERDONCKT J., *The CAMELEON Reference Framework, deliverable D1.1*, IST FP5 CAMELON project, [http://giove.isti.cnr.it/projects/cameleon/deliverable1\\_1.html](http://giove.isti.cnr.it/projects/cameleon/deliverable1_1.html)
- [CAL 03] CALVARY G., COUTAZ J., THEVENIN D., LIMBOURG Q., SOUCHON N., BOUILLON L., VANDERDONCKT J., *A Unifying Reference Framework for Multi-target User Interfaces*. *Interacting with Computers*, Elsevier Science B.V., June, 15(3), 2003, pp 289-308.
- [CAL 07] CALVARY G., *Plasticité des Interfaces Homme-Machine*, Habilitation à Diriger des Recherches, Université Joseph Fourier, Grenoble 1, Novembre 2007.
- [CAL 10] CALVARY G., SERNA A., KOLSKI C., COUTAZ J., *Les transports : un terrain fertile pour la plasticité des interfaces homme-machine*. In C. Kolski (Ed.), *Interaction homme-machine dans les transports - information voyageur, personnalisation et assistance*, Hermes Science Publications, Paris, 2010, pp. 287-312.
- [CAN 09] CANFORA G., DI PENTA M., LOMBARDI P., VILLANI M.L., *Dynamic Composition of Web Applications in Human centered Processes*. IEEE PESOS'09, May 18-19, 2009.
- [CAO 06] CAO X., BALAKRISHNAN R., *Interacting with dynamically defined information spaces using a handheld projector and a pen*, Proceedings of UIST 2006, ACM Symposium on User Interface Software and Technology, ACM Press, 2006, pp 225-234.
- [CAR 83] CARD S.K., MORAN T.P., NEWELL A., *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983.
- [CAR 84] CARROLL J.M., CARRITHERS C., *Training wheels in a user interface*, Communication of the ACM, 27(8), ACM New York, 1984.
- [CAR 00] CARROLL J.M., *Making Use, Scenario-based design of Human-Computer Interactions*, MIT Press, 2000.
- [CON 99] CONSTANTINE L.L., LOCKWOOD L.A.D., *Software for Use : A Practical Guide to the Models and Methods of Usage-Centred Design*. New-York, Addison-Wesley, 1999.
- [CON 03] CONINX K., LUYTEN K., VANDERVELPEN C., VAN DEN BERGH J., CREEMERS B., Dygimes: *Dynamically generating interfaces for mobile computing devices and embedded Systems*. In Proceedings Mobile HCI 2003, 2003, pp 256-270.
- [COU 95] COUTAZ J., NIGAY L., SALBER D., BLANDFORD A., MAY, J. & YOUNG R., *Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE properties*, Proceedings of the INTERACT'95, Chapman&Hall Publ., 1995, pp 115-120.
- [COU 05] COUTAZ J., CROWLEY J., DOBSON S., GARLAN D., *Context is key*. Communications of the ACM, ACM Publ., 48(3), 2005, pp 49-53.
- [COU 06] COUTAZ J., *Meta-User Interfaces for Ambient Spaces*, Invited speaker. Proc. of TAMODIA 2006, Springer LNCS 4385, 1-15, Hasselt, Belgium, October 2006.

[COY 04] COYETTE A., FAULKNER S., KOLP M., LIMBOURG Q., VANDERDONCKT J., *SketchiXML: towards a multi-agent design tool for sketching user interfaces based on USIXML*. In Proceedings of the 3rd Annual Conference on Task Models and Diagrams, TAMODIA 2004, Prague, Czech Republic, 2004.

[CRO 02] CROWLEY J.L., COUTAZ J., REY G., REIGNIER P., *Perceptual components for context aware computing*, in Proc. International Conference on Ubiquitous Computing (UbiComp 2002), 2002, pp 117-134.

[CRO 04] CROWLE S., HOLE L., *ISML : An Interface Specification Meta-Language*, in Proc. Interactive Systems. Design, Specification, and Verification 2004 (DSV-IS 2004), Van Leeuwen, J., Hartmanis, J. and Goos, G., eds. Berlin / Heidelberg: Springer, 2004, pp 362-376.

[DEM 05] DEMEURE A., BALME L., CALVARY G., *CamNote : A plastic slides viewer*. In Proc. Plastic Services for Mobile Devices (PSMD) workshop held in conjunction with INTERACT'05, Rome, 2005.

[DEM 07] DEMEURE A., *Modèles et outils pour la conception et l'exécution d'Interfaces Homme-Machine Plastiques*, Thèse de l'Université Joseph Fourier, Grenoble I, Octobre 2007.

[DEM 08] DEMEURE A., CALVARY G., CONINX K., *A Software Architecture Style and an Interactors Toolkit for Plastic User Interfaces*. In Proceeding of the 15th international workshop DSV-IS 2008, LNCS, Springer Berlin, 2008, pp 225-237.

[DEY 01] DEY A. K., *Understanding and using Context*. Journal of Personal and Ubiquitous Computing, Springer London, Vol 5, 2001, pp 4-7.

[DIE 93] DIETERICH H., MALINOWSKI U., KÜHME T., SCHNEIDER-HUFSCHEIDT M., *State of the art in adaptive user interfaces*, In: Adaptive User Interfaces, Principle and Practice, Human Factors in Information Technology series, Volume 10, Elsevier Science Publishers, 1993, pp 13-48.

[DIX 93] DIX A., FINLAY J., ABOARD G., BEALE R., *Human-Computer Interaction*, Prentice-Hall, Upper Saddle River, New-Jersey, United States, 1993.

[DUA 06] DUARTE C., CARRIÇO., *A conceptual framework for developing adaptive multimodal applications*. In Proc. of the 11th international conference on Intelligent User Interfaces, IUI'06, ACM, New York, 2006, pp 132-139.

[EIS 00] EISENSTEIN J., VANDERDONCKT J., PUERTA A., *Adapting to Mobile Contexts with User-Interface Modeling*, Proceedings of 3rd IEEE Workshop on Mobile Computing Systems and Applications WMCSA'2000 (Monterey, 7-8 December 2000), IEEE Press, Los Alamitos, 2000, pp 83-92.

[EUZ 08] EUZÉNAT J., PIERSON J., RAMPARANY F., *Dynamic context management for pervasive applications*, Knowledge Engineering Review, 23(1), 2003, pp 21-49.

[FER 09] FERRY N., HOURDIN G., LAVIROTTE S., REY G., TIGLI J.-Y., RIVEILL M., *Models at Runtime: Service for Device Composition and Adaptation*. In 4th International Workshop Models@run.time, Models 2009 (MRT'09), 2009.

- [FRA 10] FRANCONI J., BAILLY G., MANDRAN N., NIGAY L., *Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-Free Selection for Expert Mode*, Proc. of the Conference on Advanced Visual Interfaces (AVI), 2010.
- [FRA 92] FRANÇOIS D., PINEAU A., ZAOUÏ A., *Elasticité et Plasticité*, Hermès, 1992.
- [GAJ 04] GAJOS K., WELD D., *SUPPLE: Automatically Generating User Interfaces*, In: Proceedings of ACM International Conference on Intelligent User Interfaces, IUI 2004, pp 93-100.
- [GAJ 05] GAJOS K., WELD D., *Preference elicitation for interface optimization*. In UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology, New York, NY, USA, 2005. ACM Press, pp 173-182.
- [GAJ 08] GAJOS K., WOBROCK J., WELD D., *Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces*. In CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM, 2008, pp 1257-1266.
- [GAR 11] GARCIA FREY A., CERET E., DUPUY-CHESSA S., CALVARY G., *QUIMERA: a Quality Metamodel to Improve Design Rationale*, In Proceedings of the third ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2011), 2011.
- [GRO 02] GROLAUX D., VAN ROY P., VANDERDONCKT J., *FlexClock, a Plastic Clock written in Oz with the Qt Toolkit*, Proc. 1st International Workshop on Task Models and Diagrams for User Interface Design (TAMODIA 2002), 2002, pp 135-142.
- [GRO 07] GROLAUX D., *Transparent Migration and Adaptation in Graphical User Interface Toolkit*, In: PhD Thesis, Faculté des sciences appliquées, Université Catholique de Louvain, Septembre 2007.
- [HAL 94] HALVERSON C.A., *Distributed Cognition as a theoretical framework for HCI: Don't throw the Baby out with the bathwater - the importance of the cursor in Air Traffic Control*. Tech Report No. 94-03, Dept. of Cognitive Science, University of California, San Diego, 1994.
- [HAR 10] HARRISON C., DESNEY T., MORRIS D., *Skinput: Appropriating the Body as an Input Surface*. In Proceedings of CHI'10, the 28th international conference on human factors in computing systems, ACM, 2010, pp 453-462.
- [HEL 08] HELMS J., SCHAEFER R., LUYTEN K., VANDERDONCKT J., VERMEULEN J., ABRAMS M., *UIML Version 4.0: Committee Draft*. Available at <http://www.oasis-open.org/committees/download.php/28457/uiml-4.0-cd01.pdf>, 2008.
- [IFI 96] IFIP, *Design Principles for Interactive Software*, livre produit par l'IFIP WG 2.7 (13.4), C. Gram et G. Cockton (dir), Chapman&Hall, London, United Kingdom, 1996.
- [JOH 03] JOHNSON P. D., PAREKH, J., *Multiple Device Markup Language a Rule Approach*. SE MS Project & Thesis (SE690), DePaul University, 2003.
- [KAR 05] KARSENTY L., BOTHEREL V., *Analyse empirique de l'inter-utilisabilité d'un service multisupport Web et téléphone*, Actes des Deuxièmes Journées Francophones sur l'Ubiquité et la Mobilité (UbiMob'05), ACM Press, Grenoble, 31 mai-3 June 2005.

- [KIE 10] KIEFFER S., COYETTE A., VANDERDONCKT J., *User Interface Design by Sketching: A complexity Analysis of Widget Representations*. In Proc. Of the 2010 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, ACM Pub., 2010, pp 57-66.
- [KOS 04] KOST S., *Dynamically generated multi-modal application interfaces*, Thèse Technical University of Dresden and Leipzig UNiversity of Applied Science, 2004.
- [KUR 02] KURTEV I., BEZIVIN J., AKSIT M., *Technological spaces: An initial appraisal*. CoopIS, DOA'2002 Federated Conferences. Industrial Track, Irvine, 2002.
- [LEC 99] LECOLINET E., *A Brick Construction Game Model for Creating Graphical User Interfaces: The Ubit Toolkit*, In: Proceedings of the 7th IFIP TC13 International Conference on Human-Computer Interaction, INTERACT 1999, Edinburgh, Scotland, UK, September 1999.
- [LIM 04] LIMBOURG Q., VANDERDONCKT J., MICHOTTE B., BOUILLON, L., LOPEZ-JAQUERO, V., *UsiXML: a Language Supporting Multi-Path Development of User Interfaces*. Proceedings of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems, EHCI-DSVIS'2004, Hamburg, July 11–13, 2004.
- [LIN 08] LIN J., LANDAY J.A., *Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces*. In Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08). ACM, New York, NY, USA, pp 1313-1322.
- [LOP 04] LOPEZ-JAQUERO V., MONTERO F., MOLINA J.P., GONZALEZ P., *A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties*, in Proc. EHCI04, Hamburg, Germany, July 2004.
- [LUY 04] LUYTEN K., ABRAMS M., VANDERDONCKT J., LIMBOURG Q., *Developing User Interfaces with XML: Advances on User Interface Description Languages*, Sattelite workshop of Advanced Visual Interfaces 2004.
- [MAE 09] MAES P., MISTRY P., *Unveiling the "Sixth Sense", game-changing wearable tech*, TED 2009, Long Beach, California, United States, 2009.
- [MIS 09] MISTRY P., MAES, P., *SixthSense – A Wearable Gestural Interface*. In Proc. SIGGRAPH Asia 2009, Emerging Technologies, Yokohama, Japan, 2009.
- [MON 04] MONTERO F., VANDERDONCKT J., LOZANO M., *Quality Models for Automated Evaluation of Web Sites Usability and Accessibility*, In Proc. International Conference on Web Engineering, ICWE'2004, Munich, Germany, July 28-30 2004.
- [MYE 90] MYERS B., *Creating User Interfaces using programming by example, visual programming, and constraints*. ACM Transaction on Programming Languages and Systems (TOPLAS), 12 (2), ACM Publ., 1990, pp 143-177.
- [MYE 00] MYERS B., HUDSON S.E., PAUSCH R., *Past, present, and future of user interface software tools*. ACM Transactions on Computer-Human Interaction (TOCHI), 7(1), 2000.
- [MYE 08] MYERS B., PARK S.Y., NAKANO Y., MUELLER G., KO A. *How designers Design and Program Interactive Behaviors*. in Proc. IEEE Symposium on Visual Languages and Human Centric Computing (VL/HCC), 2008, pp 177-184.

- [NIC 06] NICHOLS J., ROTHROCK B., CHAU D. H., MYERS B.A., *Huddle: Automatically Generating Interfaces for Systems of Multiple Connected Appliances*. In: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, UIST 2006, pp 279-288.
- [NIE 94] NIELSEN J., *Heuristic evaluation*, dans Nielsen J., Mack R.L. (dir.), Usability Inspection Methods, John Wiley & Sons, New York, 1994.
- [NIG 94] NIGAY L., *Conception et modélisation logicielles des systèmes interactifs : applications aux interfaces multimodales*. Thèse de l'Université Joseph Fourier, 1994.
- [NIG 96] NIGAY L., COUTAZ J., *Espaces conceptuels pour l'interaction multimédia et multimodale*. Technique et Science Informatique (TSI), spécial Multimédia et Collecticiel, Hermès, 15(9), 1996.
- [NOG 08] NOGIER J.F., *Ergonomie du logiciel et design web - Le manuel des interfaces utilisateur*, 4e édition, Dunod, Paris, 2008.
- [NOR 79] NORMAN D.A., BOBROW D.G., *Descriptions: An intermediate stage in memory retrieval*. Cognitive Psychology, 11, 1979, pp 107-123.
- [NOR 86] NORMAN D.A., DRAPER S.W., *User Centered System Design: New Perspectives on Human-Computer Interaction*. Erlbaum Associates, Hillsdale, NJ, 1986.
- [PAT 03a] PATERNO F., *Concur Task Trees: An engineered notation for task models*. In D. Diaper, & N. Stanton, (Eds.), The Handbook of Task Analysis for Human-Computer Interaction (pp. 483–503 ch. 24). Lawrence Erlbaum Associates, 2003.
- [PAT 03b] PATERNO F., SANTORO A., *Unified Method for Designing Interactive Systems Adaptable to Mobile and Stationary Platforms*, Interacting with Computers, Elsevier, 15, 2003, pp 349-366.
- [PAT 08] PATERNO F., SANTORO C., MANTYJARVI J., MORI G., SANSONE S., *Authoring pervasive multimodal user interfaces*. International Journal of Web Engineering and Technology, 4(2), 2008.
- [PAT 09] PATERNO F., SANTORO C., SPANO L.D., *MARIA: a universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments*, in ACM Transactions on Computer-Human Interaction (TOCHI), 16 (4), ACM Press, 2009.
- [PIC 97] PICARD R.W., *Affective Computing*, MIT Press, 1997.
- [PIC 03] PICARD E., FIERSTONE J., PINNA-DERY A-M., M. RIVEILL. *Atelier de composition d'IHM et évaluation du modèle de composants*. Livrable I3, RNTL ASPECT, Laboratoire I3S, 2003.
- [PIE 09] PIETSCHMANN S., VOIGT M., MEIBNER K. *Dynamic Composition of Service-Oriented Web User Interfaces*. Proc. of the 4th International Conf. on Internet and Web Applications and Services, ICIW 2009, IEEE CPS, ISBN 9780769536132, 2009.
- [PON 01] PONNEKANTI S., LEE B., FOX A., HANRAHAN P., WINOGRAD T., *Icrafter: A service framework for ubiquitous computing environments*. In G. Abowd, B. Brumitt, S. Shafer (Eds.) Proceedings Ubicomp 2001, Springer Publ., LNCS 2201, 2001, pp 57–75.

- [PRE 94] PREECE J., ROGERS Y., SHARP H., BENYON D., HOLLAND S., CAREY T., *Human-Computer Interaction*, Addison Wesley Publication, Wokingham, United Kingdom, 1994.
- [PUE 01] PUERTA A., EISENSTEIN J. *XIML: A common representation for interaction data*. Proceedings IUI01, ACM publ., 2001, pp 214–215.
- [REI 07] REIGNIER P., BRDICZKA O., VAUFREYDAZ D., CROWLEY J.L., MAISONNASSE J. *Context Aware Environments : from Specification to Implementation*. Expert Systems: The Journal of knowledge Engineering, 5(24), 2007, pp 304-320.
- [REP 04] REPENNING A., IOANNIDOU A., *Agent-Based End-User Development*. Communications of the ACM, 47(9), 2004, pp 43-46.
- [REY 05] REY G., *Le Contexte en interaction homme-machine: le contexteur*. Thèse Université Joseph Fourier, 2005.
- [ROB 91] ROBERTSON G.G., MACKINLAY J.D., CARD S.K., *Cone Trees: Animated 3D Visualizations of Hierarchical Information*, in Proceedings of the ACM CHI 91 Human Factors in Computing Systems Conference, 1991, pp 189-194.
- [ROD 04] RODDEN T., CRABTREE A., HEMMINGS T., KOLEVA B., HUMBLE J., AKESSON K.P., HANSSON P., *Configuring the Ubiquitous Home*. In Proc of the 2004 ACM Symposium on Designing Interactive Systems (DIS 2004), Massachusetts: ACM Press, 2004.
- [ROU 03] ROUILLARD J., *Plastic ML an dits Toolkit*, in Proc HCI International, 2003.
- [ROU 06] ROUSSEAU C., BELLIK Y., VERNIER F., *A Conceptual Model for Multimodal and Contextual Presentation of Information*, In: French Human-Computer Interaction Journal, RIHM, Volume 7, 2006.
- [SCH 06] SCHAEFER R., BLEUL S., *Towards object oriented, uiml-based interface descriptions for mobile devices*. In Computer-Aided Design of User Interfaces V", Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006 (Bucharest, 6-8 June 2006)
- [SEF 04] SEFFAH A., JAVAHERY H., *Multiple user interfaces, cross-platform applications and context-aware interfaces*, Wiley & Sons, New York, United States, 2004
- [SER 09] SERRANO M., NIGAY L., *Temporal Aspects of CARE-based Multimodal Fusion: From a Fusion Mechanism to Composition Components and WoZ Components.*, In Proceedings of the 11th international conference on Multimodal interfaces (IMCI '09), 2009, pp 177-184.
- [SHA 91] SHACKEL B., *Usability-Context, Framework, Design and Evaluation*, Human Factors for Informatics Usability, Cambridge University Press, Cambridge, 1991, pp 21-38.
- [SHN 97] SCHNEIDERMAN B., *Designing User Interface Strategies for effective Human-Computer Interaction*, 3e edition, Addison-Wesley, Reading, United States, 1997.
- [SHN 11] SHNEIDERMAN B., *Technology-Mediated Social Participation: The Next 25 Years of HCI Challenges*, HCI International, 2011.
- [SHO 07] SHOEMAKER G., TANG A., BOOTH K., *Shadow reaching: a new perspective on interaction for large displays*, Proc. User Interface Software and Technology 2007 (UIST 2007), ACM Press, 2007.



- [SMI 93] SMITH D.C., *Pygmalion: An executable Electronic Blackboard*. Chapter1 In “Watch What I Do”, A. Cypher ed., The MIT Press, 1993.
- [SOT 07] SOTTET J.-S., CALVARY G., COUTAZ J., FAVRE J.-M., *A Model-Driven Engineering Approach for the Usability of User Interfaces*. In Proc. Engineering Interactive Systems (EIS2007), J. Gulliksen et al. (eds), LNCS 4940, 2007, pp 140-157.
- [SOT 08] SOTTET J.-S., *Méga-IHM : Malléabilité des Interfaces Homme-Machine dirigées par les modèles*, Thèse Université Joseph Fourier, 2008.
- [SOU 03] SOUSA J.P., GARLAN D., *The Aura Software Architecture : an Infrastructure for Ubiquitous Computing*, Carnegie Mellon Technical Report, CMU-CS-03-183, 2003.
- [SOU 05] SOUSA J.-P., *Scaling Task Management in Space and Time: Reducing User Overhead in Ubiquitous-Computing Environments*, In: Ph.D. Thesis, Carnegie Mellon University, CMU Technical Report CMU-CS-05-123, March 2005.
- [STE 09] STEPHANIDIS C., *The Universal Access Handbook*, CRC Press, 2009.
- [STR 99] STREITZ N., GEIBLER J., HOLMER T., KONOMI S., MÜLLER-TOMFELDE C., REISCHL W., REXROTH P., SEITZ P., STEINMETZ R., *i-LAND: An interactive Landscape for Creativity and Innovation*. In Proceedings. of the ACM conference On Human Factors in Computer Human Interaction (CHI99), ACM, 1999, pp 120–127.
- [STU 06] STUERZLINGER W., CHAPUIS O., PHILLIPS D., ROUSSEL N., *User interface façades: towards fully adaptable user interfaces*. In: Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology, UIST 2006, 2006, pp 309-318.
- [SUC 87] SUCHMAN L., *Plans and Situated Actions*, Cambridge University Press, 1987.
- [TAN 01] TANDLER P., *Software Infrastructure for Ubiquitous Computing Environments: Supporting Synchronous Collaboration with Heterogeneous Devices*, In: Proceedings of UBICOMP 2001, LNCS 2201, Atlanta, GA, USA, 2001, pp 96-115.
- [THE 99] THEVENIN D., COUTAZ J., *Plasticity of User Interfaces: Framework and Research Agenda.*, Proc. Interact99, Edinbourg, dans A. Sasse & C. Johnson (dir.), IFIP IOS Press Publication, 1999, pp 110-117.
- [THE 01] THEVENIN D., *L'adaptation en Interaction Homme-Machine : le cas de la plasticité*. Thèse de l'Université Joseph Fourier, 2001.
- [TRE 03] TREVISAN D., VANDERDONCKT J., MACQ B., *Continuity as a usability property*, HCI 2003 – 10th International Conference on Human-Computer Interaction, vol. I, Heraklion, Grece, June 22-27 2003, pp 1268-1272.
- [TRU 04] TRUONG K.N., HUANG E.M., ABOWD G., *CAMP: a magnetic poetry interface for end-user programming of capture applications for the home*. In Proc. of the 6th international conf. On ubiquitous computing (UbiComp'04), 2004, pp 143-160.
- [USI 11] <http://itea.defimedia.be/>
- [VAN 99] VAN WELIE M., VAN DER VEER G.C., ELIENS A., *Usability Properties in Dialog Models*, 6th International Eurographics Workshop on Design Specification and Verification of Interactive Systems DSV-IS99, Braga, Portugal, 2-4 juin 1999, pp 238-253.

50 Titre de l'ouvrage

[WIN 01] WINOGRAD T., *Architectures for Context. Human-Computer Interaction – Special Issue on Context-Aware Computing*, Lawrence Erlbaum Associates, 16 (2-4), 2001, pp 401-420.

[YU 07] YU J., BENATALLAH B., SAINT-PAUL R., CASATI F., DANIEL F. M. MATERA, M., *A Framework for rapid Integration of Presentation Components*. In WWW'07 Proc. of the 16th International Conf. on World Wide Web, 2007, 2007, pp 923-932.

## Annexe 1 : Il y a plasticité et plasticité !

Rappelons notre définition :

« La plasticité de l'interface homme-machine d'un système interactif dénote la capacité d'adaptation de cette interface au contexte d'usage pour en préserver l'utilité et l'utilisabilité, et par extension, la valeur tout en accordant à l'utilisateur les moyens de contrôle adéquats ».

Nous reformulons ainsi la propriété d'être plastique en termes plus précis :

« Une IHM est dite plastique pour une valeur  $V$  sur un ensemble de contextes d'usage  $C$  si elle est capable de s'adapter à tout changement de contexte d'usage sur  $C$  dans le respect de  $V$  ».

La définition de la plasticité donne lieu à différentes qualifications et quantifications en jouant sur  $V$  et sur  $C$ . D'un point de vue qualitatif, nous nuancions selon l'exactitude de la valeur  $V$  et des contextes d'usage  $C$ . La plasticité est dite [CAL 07] :

- maximale en valeur (versus minimale) s'il n'existe pas de propriété  $p$  non considérée dans  $V$  telle que l'IHM soit plastique pour  $V \cup \{p\}$  sur  $C$  ;
- maximale en contexte d'usage (versus minimale) s'il n'existe pas de contexte d'usage  $c$  non considéré dans  $C$  tel que l'IHM soit plastique pour  $V$  sur  $C \cup \{c\}$  ;
- maximale (versus minimale) si la plasticité est maximale en valeur et en contexte d'usage. Si, à l'usage, la valeur  $V$  ou les contextes d'usage  $C$  augmentent par apprentissage, la plasticité est dite incrémentale. S'ils diminuent, elle est dite décrémentationale. S'ils varient, elle est dite élastique. S'ils sont stables, elle est dite rigide.

D'un point de vue quantitatif, le domaine de plasticité est une mesure possible de la plasticité [CAL 01]. Pour une valeur donnée  $V$ , nous appelons îlot de plasticité un ensemble de contextes d'usage  $C$  tel que l'IHM est maximale ment plastique en contexte d'usage pour  $V$  sur  $C$ . Le domaine de plasticité de l'IHM pour la valeur  $V$  est l'union des îlots de plasticité de cette IHM pour cette valeur  $V$ . La figure 9.13 montre quatre îlots de plasticité. Le domaine de plasticité est l'union de ces îlots.

## Annexe 2 : Outils de mise en œuvre

L'analyse de l'état de l'art sur la plasticité des IHM révèle une grande diversité d'approches, chacune d'elles véhiculant des progrès sensibles sans qu'aucune ne couvre tous les aspects de l'espace problème. On relève néanmoins deux approches complémentaires : l'une, dirigée par les modèles, est inspirée de l'IDM ; l'autre, centrée sur le développement de briques logicielles réutilisables, est plutôt orientée code sous forme de gestionnaires de fenêtres, de boîtes à outils et d'intergiciels, ces derniers pouvant utilement s'inspirer des approches à composants orientés services. Bien que ces deux grandes familles d'approches soient adoptées par des communautés de recherche qui s'ignorent, elles sont fondamentalement conciliables.

### *Approches dirigées par les modèles*

L'approche dirigée par les modèles convient aux domaines métier pour lesquels il existe une réelle capitalisation des savoir-faire en matière d'IHM. Cette capitalisation s'exprime sous forme de patrons mis en pratique par des assemblages d'interacteurs WIMP (Windows Icons Menus Pointers) de référence. Des boîtes à outils post-WIMP commencent à apparaître mais leur intégration dans les approches génératives n'a pas été envisagée. Comme le montre le Tableau 9.1, un grand

nombre de langages de description prend l'IHM-A comme point d'entrée. Nous avons exclu de la table tous les langages de description d'IHM-C comme XUL, MXML, EMMA, inkML, VRML, VoiceXML et bien d'autres ! Très peu de langages partent du modèle de tâche. Tous sont concernés par le remodelage mais aucun ne traite la redistribution ni la qualité (à l'exception de UsiXML dans ses versions futures [USI 11] [GAR 11]). À notre connaissance, les outils qui utilisent ces modèles pour générer des IHM concrètes et finales perdent le mapping avec les modèles sources. En conséquence, à l'exécution, l'IHM ne peut qu'être régénérée impliquant un déploiement statique avec reprise en début de session. En l'état, l'approche centrée code est incontournable.

### *Approche orientée code*

Concernant l'aspect « code », on constate l'absence de gestionnaire de ressources d'interaction adapté aux conditions de l'informatique ambiante (la virtualisation des ressources d'interaction serait ici bienvenue !). En conséquence, les développeurs de boîtes à outils comme Ubit [LEC 99] ou Etk [GRO 07] tentent de pallier ce manque mais au sein d'un espace technologique unique. Les COMETs font exception [DEM 07]. Le Tableau 9.2 montre en synthèse l'offre de l'état de l'art en matière de window manager et de boîtes à outils. Si ces dernières étaient couplées aux outils de génération fondés sur les langages du Tableau 9.2, nous gagnerions en richesse d'interaction.

Nous relevons deux classes d'infrastructures selon que la classe de problèmes traités concerne la redistribution (BEACH [TAN 01], Aura [SOU 05]), le remodelage (ICrafter [PON 01], Eloquence [ROU 06]) et l'agrégation d'IHM - un remodelage particulier dont l'importance va croissant (Huddle [NIC 06]). Comme le montre le Tableau 9.3, aucune de ces solutions, à l'exception d'Ethylène [BAL 08], ne couvre à la fois le remodelage et la redistribution. La raison tient essentiellement aux domaines d'application qui définissent de facto le contour de la solution : BEACH, ICrafter, Huddle traitent de milieux confinés relativement stables et contrôlables ; Aura introduit un concept nouveau (celui d'espace informationnel qui suit l'utilisateur en tout lieu), mais s'appuie sur des applications patrimoniales traditionnelles. Ethylène vise le problème général de la plasticité en appliquant CAMELEON-RT.

Approches centrées IDM	Remarques	Couverture des modèles de CAMELEON/Espace problème
<b>Langages</b>		
XIML [PUE 01], DISL [SCH 06], GIML [KOS 04], MDML [JOH 03], PlasticML [ROU 03], SunML [PIC 03]	DISL : support multimodalité, une extension de UIML [ABR 04] [HEL 08] ; SunML : opérations de composition de widgets ;	Remodelage total de toute l'IHM. Contexte d'usage couvert : plate-forme élémentaire. Modèles couverts : <u>IHM-A avec dialogue (comportement)</u> . Génération d'IHM-F par <u>réification vers un seul espace technologique</u> , mais <u>choix entre plusieurs espaces technologiques cibles</u> (par ex., DHTML, WML, VoiceXML, Java, C++, Perl).
<u>ISML</u> [CRO 04], <u>TeresaXML</u> [PAT 03b], <u>MariaXML</u> [PAT 09], <u>XIML</u> [EIS 00]	MariaXML (outil Maria qui fait suite à Teresa) : support pour la multimodalité avec Complémentarité et Redondance.	Remodelage total de toute l'IHM. Contexte d'usage couvert : plate-forme élémentaire. Modèles couverts : <u>Modèle de Tâche, modèle des concepts métier</u> . Génération d'IHM-F par <u>réification vers un seul espace technologique</u> (le plus souvent Java et ML du monde Web - WML, XHTML, VoiceXML).
<u>SeescoaXML</u> [LUY 04]	SeescoaXML (outil Seescoa) : génération à l'exécution vers plusieurs plates-formes cibles élémentaires. Dygimes [CON 03], une extension de Seescoa [LUY 04] part d'un modèle de tâche CTT traduit ensuite en IHM-A.	Remodelage total de toute l'IHM. Contexte d'usage couvert : plate-forme élémentaire. Modèles couverts : <u>IHM-A</u> . Génération d'IHM-F <u>vers plusieurs espaces technologiques simultanément</u> (par ex., HTML, Java AWT, Java Swing), d'abord par traduction d'IHM-A, puis pour chaque IHM-A, par réification avec XSLT.
<u>UsiXML</u> [LIM 04]. Vise à couvrir tous les modèles et principes du cadre de référence CAMELON original et de là, servir de norme.	Existence d'outils interopérables et d'infrastructures run-time compatibles UsiXML	Couverture complète des modèles CAMELEON. Tout l'espace problème est <u>potentiellement</u> couvert sous réserve que les limitations de l'IDM soient levées (cf. section 9.6) et que les outils et infrastructures d'exécution idoines soient développés.
Damask [LIN 08]. Vise à couvrir les premières phases de conception.	Damask permet de manipuler des sketches et prend en compte la conception d'interfaces vocales.	Couverture du niveau interface concrète. Damask permet au concepteur d'assembler des patterns.

**Tableau 9.1 :** Exemples de langages de description d'IHM. Les langages de description d'IHM concrètes ne sont pas mentionnés.

Approches centrées Code (1/2)	Hypothèses	Couverture de l'espace problème
<i>Gestionnaire de fenêtres</i>		
FACADE [STU 06]. IHM d'applications patrimoniales reconstruites à façon par drag&drop par l'utilisateur (façon mash-up)	IHM patrimoniales développées avec des boîtes à outils avec capacité d'instrospection. Existence du renderer Métisse au-dessus de X Window.	Méta-IHM EUD, remodelage de niveau interacteur, déploiement dynamique avec reprise au niveau de l'action. Remodelage intra-modal graphique niveau PL et PP, contexte d'usage = plate-forme élémentaire ou composite statique équipée de Métisse.
<i>Boîtes à outils</i>		
Ubit [LEC 99]. Rendu multiple d'un même widget (graphe de scène et nœuds condition).	Au programmeur de prévoir toutes les formes de rendu.	Mécanismes de base pour le remodelage dynamique intra-modal graphique d'interacteur au niveau de l'action physique. Contexte d'usage = plate-forme élémentaire ou composite statique.
Multimodal Widget [CRE 00]. Rendu multiple d'un widget dans plusieurs modalités en fonction des ressources d'interaction. Par exemple, un bouton graphique, à la visite de la souris, peut à la fois se repeindre en jaune et émettre un son si un générateur de son est disponible.	Au programmeur de prévoir toutes les formes de rendu.	Mécanismes de base pour le remodelage dynamique inter-modal d'interacteur au niveau de l'action physique. Contexte d'usage = plate-forme élémentaire aux ressources d'interaction dynamiques. Méta-IHM pour contrôler la proportion de modalités souhaitées (ex., 100% graphique et 30% sonore).
ETk [GRO 07]. Widgets usuels de Tk mais découpables, remplaçables, recomposables et migrables.	Existence d'un unique espace technologique qui masque l'hétérogénéité des plates-formes qui peuvent se composer dynamiquement : Mozart [VAN 04]	Couvre presque tout l'espace problème mais pour un unique espace technologique : le système Mozart, son langage Oz et Tk, dont il exploite les propriétés pour la redistribution et le remodelage.
COMET [DEM 07]. Arborescence de widgets (les COMETs) isomorphe à un modèle de tâche. Certaines COMETs jouent le rôle d'opérateurs de tâches, les autres couvrent chacune une sous-tâche, avec plusieurs IHM-A où chaque IHM-A peut avoir plusieurs IHM-C et IHM-F implémentées dans des espaces technologiques distincts.	Au programmeur de prévoir toutes les formes de rendu à l'aide de feuilles de styles s'appuyant sur le polymorphisme des COMETs.	Mécanismes de base pour le remodelage dynamique intra- et inter-modal à l'aide de feuilles de styles. Redistribution de l'IHM via l'implémentation HTML ou B207. Contexte d'usage = plate-forme élémentaire ou composite dynamique.

Tableau 9.2 : Exemples de boîtes à outils exploitables pour des IHM plastiques.

Approches centrées Code (2/2)	Hypothèses/Cible	Couverture de l'espace problème
<b>Infrastructures et redistribution d'IHM</b>		
<p><u>Beach</u> [TAN 01]. L'un des premiers intergiciel pour "RoomWare" collaboratif (Projet i-LAND [STR 99]).</p>	<p>Les stations de travail de la plate-forme composite sont des PC dotés de ressources d'interaction homogènes. Par ex., le Dynawall = plusieurs écrans "SmartBoard" de taille et de résolution identiques.</p>	<p>Absence de remodelage. Redistribution des IHM au niveau de l'interacteur (voire du pixel), avec reprise au niveau de l'action physique et redéploiement dynamique. Pour le contexte d'usage, absence d'adaptation à l'utilisateur, mais plate-forme dynamique homogène. Un seul espace technologique (celui de Beach) et méta-IHM avec négociation pour commander la redistribution de l'IHM.</p>
<p><u>Aura</u> [SOU 03] [SOU 05]. Halo computationnel attaché à l'utilisateur en toute circonstance : vise la continuité de services lorsque l'utilisateur se déplace d'un lieu à l'autre. Par exemple, l'utilisateur commence l'édition d'un document avec MS Word puis se rend en un nouvel endroit où MS Word n'est pas disponible, il est remplacé par un autre éditeur avec le document ouvert au bon endroit. Aura choisit dynamiquement les services les mieux adaptés aux préférences de l'utilisateur, préférences spécifiées par une méta-IHM (Aura n'apprend pas un modèle de l'utilisateur).</p>	<p>Les applications patrimoniales sont ouvertes. Elles doivent offrir les "bonnes API" d'inspection en sorte de modéliser l'état de ces applications pour assurer la continuité de service.</p>	<p>Redistribution de granularité « total » (l'unité de migration correspond à une, voire à plusieurs, applications). S'il y a redistribution fine de l'IHM, remodelage ou multimodalité, c'est uniquement le fait des applications natives. Etat de reprise dépendant de la finesse des inspections (API) permises par les applications (<i>a minima</i>, niveau tâche). Contexte d'usage : plate-forme (mais pour les services logiciels seulement), utilisateur (en termes d'expression des préférences) et environnement physique. <u>Aura est clairement inter-espaces technologiques</u>, mais à gros-grain. Méta-IHM EUP élémentaire.</p>
<b>Infrastructures et remodelage d'IHM</b>		
<p><u>Eloquence</u> [ROU 06]. Ensemble d'outils pour concevoir, générer et exécuter des IHM-F multimodales en sortie capables de remodelage à l'exécution. Le choix de la modalité de sortie dépend de la disponibilité (dynamique) des "renderers" de modalités.</p>	<p>Règles d'adaptation spécifiées par le concepteur d'IHM. Par exemple, si l'unité informationnelle est un appel téléphonique et si la batterie est faible, alors exclure la modalité photo de l'appelant et le vibreur. Plate-forme élémentaire.</p>	<p>Absence de redistribution. <u>Remodelage multimodal avec support des propriétés CARE de niveau interacteur</u> (mais en sortie seulement), avec reprise au niveau de l'action physique. Adaptation dynamique aux ressources d'interaction de la plate-forme (élémentaire) - Incertitude pour les autres aspects du contexte d'usage. Absence de méta-IHM (règles d'adaptation pré-codées par le concepteur). Probablement support inter-espaces technologiques (au mieux).</p>



<p><b>Huddle</b> [NIC 06]. Configuration d'appareils multimédia personnels. Modèle flux de données construit par l'utilisateur via une méta-IHM. Par exemple, connecter le flux image en provenance de l'antenne TV à l'écran de télévision et relier le flux audio de l'antenne aux haut-parleurs de la chaîne HI-FI. Génération d'IHM agrégées.</p>	<p>Plate-forme cible limitée à des appareils multimédia personnels. Génération d'IHM de contrôle de ces appareils sur PDA. Les appareils doivent être méta-décrits dans un langage (XML) dédié.</p>	<p>Absence de redistribution. Remodelage intra-modal graphique conventionnel, de niveau total avec reprise au niveau session, mais l'IHM est régénérée dynamiquement. Contexte d'usage : adaptation à la plate-forme qui comprend un PDA de rendu de l'IHM et les appareils multimédia. L'espace technologique couvert est intra-ET (celui de Huddle). Méta-IHM EUP pour construire les flux d'information entre les appareils.</p>
<p><b>ICrafter</b> [PON 01]. Salles interactives, déploiement de nouveaux services et agrégation de services où un service désigne une application (un navigateur Web, MS PowerPoint) ou des dispositifs utilitaires physiques (lampes, vidéoprojecteur). Modèle flux de données. Une méta-IHM permet à l'utilisateur de choisir les services souhaités. ICrafter lui retourne une IHM agrégat (graphique conventionnelle), union des IHM des services choisis.</p>	<p>Les services qui incluent leur propre IHM (tel MS Word) échappent au remodelage. Le modèle du contexte est exprimé dans un langage XML dédié : description de la plate-forme cible (langages et interprètes comme html et navigateur web), configuration de la salle interactive (relations spatiales entre dispositifs utilitaires, etc.).</p>	<p>Remodelage, par génération d'IHM, pour des services et de composition de services ne disposant pas d'IHM. La granularité de l'état de reprise est celui de la session, mais le déploiement d'IHM est commandé dynamiquement sous le contrôle de l'utilisateur grâce à une méta-IHM de négociation générée de la même façon que les autres IHM (méta-IHM remodelable) avec adaptation à l'environnement physique et à la plate-forme élémentaire cible. ICrafter est de niveau inter-espaces technologiques : les plates-formes élémentaires d'une salle ICrafter pouvant offrir des espaces technologiques différents, les IHM générées via ICrafter peuvent être exprimées dans des espaces technologiques différents.</p>
<p><b>Ethylene</b> [BAL 08]. Approche hybride IDM et composants orientés services. Infrastructure conforme à CAMELEON-RT. Utilisée pour la mise en œuvre de PhotoBrowser (Figure 9.6). Applique les principes de la section 9.6</p>	<p>Chaque plate-forme élémentaire doit être capable d'exécuter Ethylene. Repose sur l'existence d'un gestionnaire de contexte (capture) et d'un annuaire de composants. Les composants doivent être métadécrits avec EthyleneXML.</p>	<p>Tout l'espace problème est <u>potentiellement</u> couvert sous réserve que toutes les fonctions de CAMELEON-RT soient implémentées.</p>

**Tableau 9.3 :** Exemples d'infrastructures exploitables pour des IHM plastiques.