

Manuscrit présenté par

Gaëlle CALVARY

Pour obtenir le diplôme d’

Habilitation à Diriger des Recherches

Spécialité Informatique

de l’Université Joseph Fourier – Grenoble I

Plasticité des Interfaces Homme-Machine

Date de soutenance : 9 Novembre 2007

Jury :

Président : Brigitte Plateau, Institut National Polytechnique de Grenoble

Rapporteurs : Michel Beaudouin-Lafon, Université Paris-Sud

Phil Gray, Université de Glasgow

Jean Vanderdonckt, Université catholique de Louvain

Examineurs : Joëlle Coutaz, Université Joseph Fourier

Jocelyne Nanard, Université de Montpellier

Philippe Palanque, Université de Toulouse

Travaux menés au sein du Laboratoire d’Informatique de Grenoble (LIG)
dans l’équipe d’Ingénierie de l’Interaction Homme-Machine (IIHM)

*A mes proches
dont la « plasticité » ne fait plus doute*

Sommaire

Remerciements	6
Préambule	8
Introduction : La plasticité, un sujet en trois temps	10
1. Problème et sujet.....	10
2. Objectifs	15
3. Démarche	15
Chapitre 1. Un espace problème pour comprendre.....	18
1. Définitions.....	18
2. Espace problème Hémisphères	22
2.1 Hémisphère droit : Directive d'adaptation.....	22
2.2 Hémisphère gauche : Cycle de vie.....	35
2.3 Hémisphères, un espace problème fédérateur.....	42
Chapitre 2. Une approche scientifique pour agir	46
1. Principes directeurs de MUST : Unification et Mixité	48
1.1 Unification.....	48
1.2 Mixité	51
2. Concepts fondateurs de MUST : Systèmes et Transformations	52
3. MUST en dix principes	54
3.1 Cas d'étude.....	54
3.2 Les dix principes de MUST	56
Chapitre 3. Des axes de recherche pour construire.....	66
1. Axes de travail actuels.....	66
1.1 Ingénierie Dirigée par les Modèles.....	66
1.2 Boîte à outils d'interacteurs pour la plasticité.....	68
1.3 Directives d'adaptation.....	71
1.4 Infrastructure logicielle pour la capitalisation	72
2. Axes de recherche naissants.....	73
2.1 Composition d'IHM par Planification Multi-Agent.....	73
2.2 Méga-IHM.....	74
2.3 Variabilité de la Valeur	74
Conclusion et perspectives : Deux temps pour l'avenir	76
Acronymes.....	80
Glossaire	82
Références	86
Annexe A : Curriculum vitae	
Annexe B : Sélection d'articles de recherche	

Remerciements

Mes remerciements s'adressent en tout premier lieu à **Joëlle** dont la mémoire, contrairement à ce qu'elle dit, ne faillit pas ☺. Trois ans après mon DEA, Joëlle ne m'a pas oubliée et accepte de me reprendre en thèse. Merci Joëlle ! Je n'oublie pas non plus **Jim** qui a participé au grand OUI. Je rejoins l'équipe en septembre 1999 pour un poste de Maître de Conférences en janvier 2000. Et depuis, je savoure chaque instant, bien consciente de la chance qui m'a été donnée. Vivre avec Joëlle, c'est apprendre chaque jour en sciences et en valeurs humaines. En voilà un Modèle, un vrai et « vivant à l'exécution » ! ☺

Je n'oublie pas non plus **Jean-Luc Voirin** et **Patrick Kijewski** de THALES qui m'ont permis de réaliser une thèse. Quitter le laboratoire et la région fut difficile, mais l'aspiration était plus forte. J'espère qu'ils l'ont compris. En tout cas, je ne les ai pas oubliés et les en remercie encore.

Je remercie chaleureusement les membres de mon jury : **Brigitte** pour avoir accepté de le présider ; **Jean, Michel** et **Phil**, mes rapporteurs ; **Jocelyne** et **Philippe**, mes examinateurs, tous deux rapporteurs de ma thèse en 1998. Michel m'a honorée de l'organisation des cours à IHM'07 ; Phil du poste de secrétaire du groupe IFIP WG2.7. Quant à Jean, c'est une longue histoire. Grâce au projet européen CAMELEON, j'ai eu l'immense plaisir de collaborer avec ce grand monsieur aperçu à IHM'95 pour la première fois. Il y aura ensuite le réseau d'excellence SIMILAR. TAMODIA'02 et CADUI'06 resteront des temps forts : la rencontre avec Teodora et la péritonite d'Alex.

J'associe à ces remerciements toute l'équipe IIBM, plus particulièrement les plasticiens qui, par leurs esprits vifs et généreux, ont donné corps à la Plasticité. Je n'oublie pas non plus la troupe **MAPPING** : « Avec MAPPING, tout est possible ». Je pense en particulier à **Jean-Marie** dont l'association nous a donné JS, **Humbert** pour l'Intelligence (Artificielle ☺) qu'il apporte à la plasticité et **Patrick** qui aura également reposé l'ancre à Grenoble après une iodisation à Brest. J'associe à ces remerciements mes amis du **LIG**, le groupe **CESAME** et tous nos collaborateurs dans les projets.

Je n'oublie pas non plus **Serge Cecconi**, dessinateur sur mesure ☺. Merci à **Didier Retour** et **Michèle Vuillet** d'avoir créé ce quatuor. Je remercie aussi **Madame André François** pour l'accord qu'elle m'a donné quant à l'utilisation de l'œuvre « Les trois temps pour bien faire ». André François était dessinateur, affichiste, peintre et sculpteur. Il avait amicalement réalisé ce dessin en 1981 pour l'album du même nom de Marcel Azzola, Patrice Caratini et Marc Fosset. Je remercie Thierry Giard de l'Association Culture Jazz, Maria Séguinel des éditions Le Chant du Monde et Patrice Caratini qui m'ont permis de remonter à Madame François.

Enfin, je pense beaucoup à mes parents, mes filles, mon mari patientissime, mon cher frère convoyeur du train de nuit Grenoble-Quimper, expert du bistouri même en châteaux de sable, ma sœur et toute sa petite tribu dont ma filleule Marion. Je terminerai par Angèle, ma petite dernière, dont la présence m'a été précieuse pendant la rédaction de ce mémoire.



Hermance



Clervie



Angèle

Préambule

Lorsque je quitte THALES (anciennement Thomson-CSF, division Radars et Contre-Mesures, puis DETEXIS) en 1999, le problème qui émergeait était celui du multi-plates-formes : les opérateurs pouvaient utiliser des consoles variées. Comment intégrer cette diversité dans la conception ? La solution "facile" mais coûteuse adoptée dans l'urgence pour un démonstrateur de surveillance maritime fut la refonte de l'Interface Homme-Machine (IHM) pour une deuxième gamme de plates-formes. Parallèlement, à Grenoble, le multi-plates-formes émerge sous un thème plus large : la Plasticité des IHM. La tentation est alors forte de rejoindre l'équipe, quittée 8 ans plus tôt puis retrouvée pendant 3 ans et demi de thèse. Je rejoins Joëlle en septembre 1999 et nous travaillons depuis de concert sur ce thème passionnant qu'est la plasticité des IHM.

La plasticité devient mon thème de recherche. Je m'attache à le rendre fédérateur de l'ensemble de mes travaux :

- mon DEA en 1991 avec James Crowley dont la collaboration a nourri le projet IMAG MAPPING pour la perception artificielle du contexte d'usage ;
- ma thèse en 1998 dans laquelle le principe de liens élastiques entre modèles et systèmes était déjà posé. Il retrouve ici un souffle nouveau, solidifié par les récentes avancées en Ingénierie Dirigée par les Modèles (IDM ou MDE pour Model Driven Engineering) ;
- mes années de pratique à THALES de 1991 à 1999 dans lesquelles s'ancre bien entendu le sujet. Elles se prolongent aujourd'hui en un projet européen commun (EMODE) dans lequel je retrouve, comme cas d'étude, le démonstrateur de surveillance maritime ☺;
- et enfin mon enseignement, notamment en ergonomie des logiciels, qui m'a permis de mieux comprendre les critères d'ergonomie et leur ancrage en conception. Le travail était ainsi préparé pour leur intégration en plasticité. La plasticité est aujourd'hui enseignée en Master II professionnel et recherche. « *Enseigner, c'est apprendre deux fois* » disait Joubert. La formation par la recherche m'aura aussi été bénéfique.

Ce rapport établit une feuille de route sur laquelle j'explique et positionne les acquis, l'approche et le futur. Chaque étudiant y a posé une pierre précieuse que je m'attacherai à situer dans l'édifice. Joëlle est évidemment en filigrane. Elle a lancé l'idée de la plasticité en 1998. Nous en sommes tous les artisans.

Introduction : La plasticité, un sujet en trois temps



Image adaptée de Marcel Azzola / Patrice Caratini / Marc Fosset, 3 temps pour bien faire.

Nos travaux se situent dans le domaine de l'Ingénierie de l'Interaction Homme-Machine. Ils visent à inventer, réaliser et évaluer de nouvelles formes d'interaction. Le sujet est la *Plasticité des Interfaces Homme-Machine (IHM)*. Nous le présentons ici ainsi que le problème auquel il répond. Nous énonçons nos objectifs et indiquons notre démarche.

1. Problème et sujet

La propriété de *Plasticité* a été introduite en 1998 et publiée en 1999 comme étant la « capacité d'une IHM à s'adapter à son contexte d'usage dans le respect de son utilisabilité » [Thevenin 99]. Dans cette définition, le contexte d'usage faisait référence à la plate-forme de l'utilisateur et son environnement physique/social. Les travaux s'ancraient dans l'informatique ambiante justifiant la propriété de plasticité par :

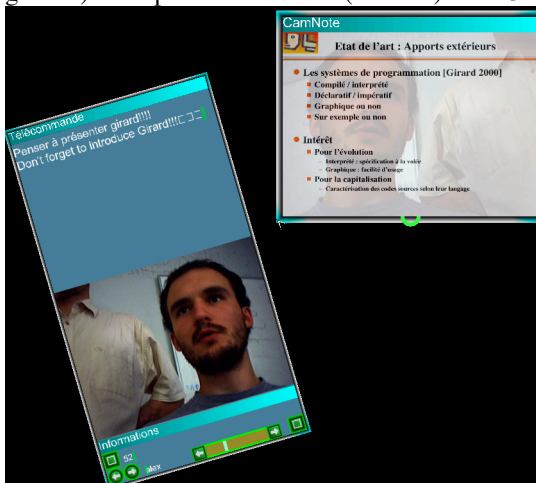
- les coûts de développement et de maintenance induits par la production d'autant de versions d'IHM que de contextes d'usage pressentis à la conception ;
- la difficulté d'assurer une cohérence ergonomique entre versions lorsque les développements sont menés de façon cloisonnée.

Cet argumentaire répondait au caractère *varié* du contexte d'usage selon une perspective système : celle du concepteur confronté à l'ingénierie d'IHM multicibles [Thevenin 01]. Nous étiquetons ces travaux comme le passage *du monociblage au multiciblage*.

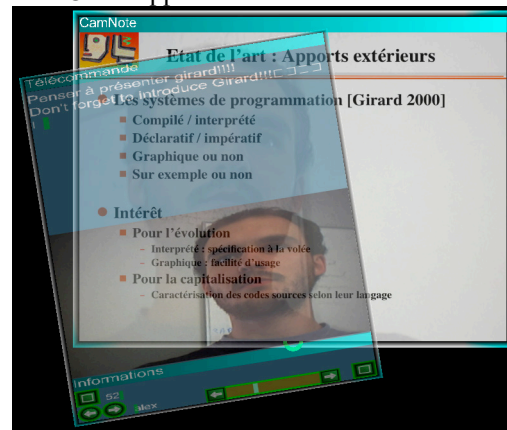
Dans un deuxième temps, le caractère *variable* du contexte d'usage est intégré à la réflexion, faisant alors le saut *du multiciblage à la plasticité* : concevoir pour plusieurs contextes d'usage n'est pas suffisant, encore faut-il assurer les changements de contexte d'usage. Le projet européen CAMELEON (2001-2004) [url CAMELEON] couvre ces deux incréments (du monociblage à la plasticité) avec un barycentre toutefois porté sur le multiciblage. En particulier, la notion d'utilisabilité, revendiquée dans la définition de la plasticité, n'est pas traitée. Les contextes d'usage et changements de contexte d'usage sont pressentis à la conception et les IHM préfabriquées. Citons à titre d'exemples nos deux démonstrateurs CamNote et Sedan-Bouillon :

- CamNote est un visualisateur de diapositives. Il permet à un présentateur de contrôler son diaporama soit en version centralisée à partir d'un PC (Figure 1a), soit en version distribuée en détachant une télécommande sur un assistant personnel (PDA) (Figure 1c). Dans le démonstrateur, le détachement ne se fait pas sur détection de l'arrivée du PDA. Il est déclenché suite à l'agrandissement, par l'utilisateur, de la fenêtre contenant la diapositive courante. La télécommande se fond alors progressivement dans la diapositive (Figure 1b) pour finalement disparaître sur PC et réapparaître remodelée sur PDA (Figure 1c). Inversement, en réduisant la diapositive, l'utilisateur rapatrie la télécommande sur PC. En terme de plasticité, CamNote illustre la *redistribution* d'IHM avec accompagnement de l'utilisateur dans le changement : la télécommande *migre* du PC vers le PDA (et vice versa) avec une *IHM de transition* [Barralon 02] (Figure 1b). CamNote illustre aussi le *remodelage*, la télécommande étant "transformée" au changement de plate-forme pour s'accommoder des ressources locales : sur PDA, la zone de commentaires et la vidéo de l'orateur (Figure 1a) disparaissent. Subsistent simplement les boutons de contrôle du diaporama (précédent, suivant, aller à la diapositive numéro x) ainsi que le réglage du niveau de transparence de la vidéo incrustée dans les diapositives (Figure 1c). L'organisation de ces éléments et leur présentation diffèrent.

a) IHM centralisée sur PC : télécommande (à gauche) et diapositive courante (à droite) sur PC



b) IHM de transition : la télécommande se fond dans la diapositive avant de disparaître sur PC et d'apparaître sur PDA



c) IHM distribuée : Diapositive sur PC (à gauche) et télécommande remodelée sur PDA (à droite)

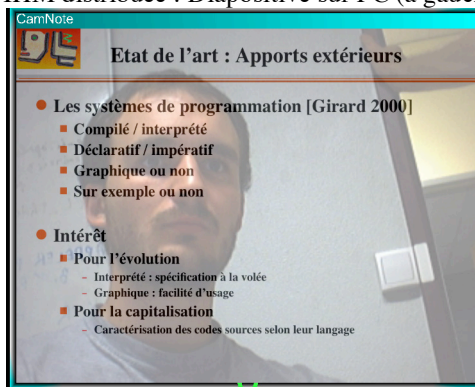
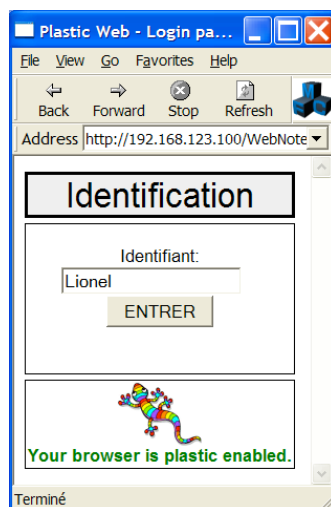


Figure 1 : CamNote, un visualisateur plastique de diapositives (A. Demeure et L. Balme).

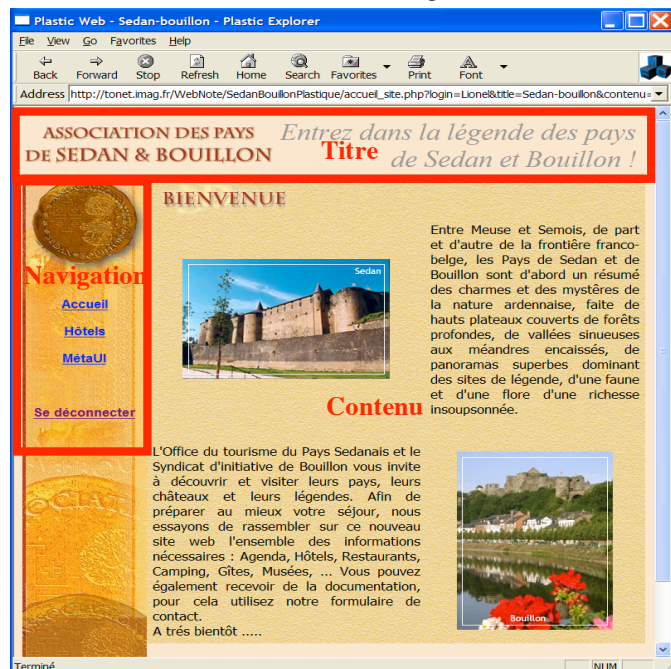
- Sedan-Bouillon est un site Web pour la promotion des pays de Sedan et Bouillon. La version plastique est une version élaborée à partir d'une version simplifiée du site

public (<http://www.bouillon-sedan.com/>). Elle est structurée en trois zones respectivement dédiées au titre, à la navigation et au contenu (Figure 2b). Dans sa version plastique, Sedan-Bouillon requiert l'identification de l'utilisateur (Lionel) (Figure 2a). Si Lionel se connecte à ce même site à partir d'un PDA, alors cette double connexion est détectée et une proposition de redistribution lui est faite : une IHM de contrôle utilisateur apparaît, expliquant à Lionel qu'il peut répartir le site, à son gré, entre le PC et le PDA (Figure 2c). La redistribution se fait au grain des espaces de dialogue (les zones). Lionel choisit, par un jeu de cases à cocher, sur quelle(s) plate(s)-forme(s) il souhaite placer ces trois espaces. La redistribution s'opère alors, composant l'IHM à façon.

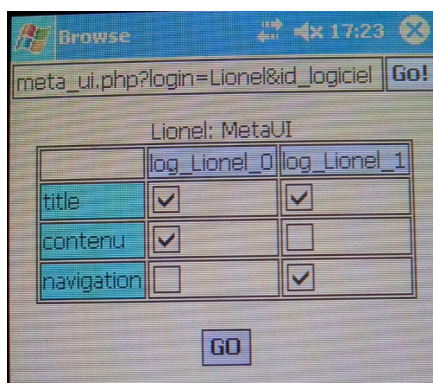
a) Connexion de Lionel sur PC



b) Sedan-Bouillon en version centralisée sur PC : l'IHM est structurée en un titre, une barre de navigation et un contenu



c) IHM de contrôle proposant la redistribution de l'IHM entre le PC et le PDA (identifiants log_Lionel_0/1)



d) Après redistribution, la navigation sur PC a disparu (cf c). Elle apparaît remodelée sur PDA : elle s'affiche horizontalement (au lieu de verticalement) en dessous du titre

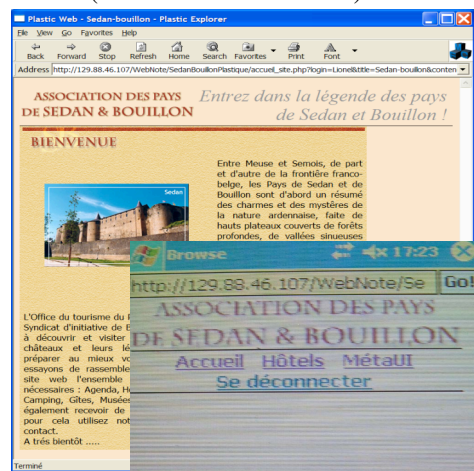


Figure 2 : Sedan-Bouillon, un site web plastique (L. Balme et A. Demeure).

Dans ces deux démonstrateurs, les plates-formes ainsi que leurs combinaisons étaient prévues à la conception. C'est dans un troisième temps que le caractère *imprévisible* du contexte d'usage est intégré à la réflexion.

L'imprévisibilité du contexte d'usage impose au système d'être capable de raisonner sur sa propre conception à l'exécution : il ne s'agit plus seulement de percevoir le contexte d'usage et de commuter vers l'IHM préfabriquée la plus appropriée, mais, si nécessaire, d'élaborer une IHM conforme aux besoins de l'utilisateur et compatible du contexte d'usage courant. Dès lors, la fonction d'adaptation prend de l'ampleur. Elle pose des problèmes d'ordre algorithmique (notamment, la composition d'IHM) et de l'ordre de l'interaction : l'adaptation doit-elle être placée sous le contrôle de l'utilisateur ?

[Coutaz 06] introduit la notion de *Méta-IHM* pour transposer, en informatique ambiante, la notion actuelle de "desktop". La Méta-IHM est un "desktop++" permettant à l'utilisateur de contrôler son espace ambiant. Elle étend le champ de perception et d'action de l'utilisateur au-delà de la station de travail conventionnelle. L'« Hyperdragging » dans les surfaces augmentées [Rekimoto 99] en est un exemple (Figure 3) : l'utilisateur sort du traditionnel écran des documents numériques qu'il visualise et déplace sur une table physique. Des faisceaux lumineux tracés sur la table le guident dans son interaction. Dans cet exemple, c'est l'utilisateur qui prend l'initiative de redistribuer son système interactif dans son espace ambiant.



Figure 3 : Un exemple de Méta-IHM dans les surfaces augmentées [Rekimoto 99].

Lorsque la redistribution est motivée par un changement de contexte d'usage comme la connexion d'une plate-forme dans Sedan-Bouillon, alors l'IHM (Figure 2c) relève de l'adaptation. Nous appelons *Extra-IHM* [Sottet 07a] cette IHM pour refléter son caractère additionnel par rapport à la seule couverture métier de l'IHM traditionnelle. L'Extra-IHM étend l'IHM au « contexte » d'usage dans son acception latine, à savoir : tout ce qui va avec le « texte » [Winograd 01] et qui est susceptible de l'influencer [Rey 05]. Le texte est ici le métier de l'utilisateur et le contexte n'est pas limité à la seule aura physique naturellement perceptible par l'utilisateur. Il peut inclure des plates-formes, pertinentes pour l'utilisateur, mais situées au-delà de son champ de vision. L'Extra-IHM peut mettre à profit des fonctions offertes par le "desktop++" comme, par exemple, la redistribution de services interactifs sur un ensemble de plates-formes. Dans le contexte de notre travail, nous réservons l'appellation Méta-IHM aux IHM de manipulation des langages de définition d'IHM, comme pratiqué en Ingénierie Dirigée par les Modèles (IDM).

Les deux démonstrateurs CamNote et Sedan-Bouillon mettent en œuvre une Extra-IHM :

- Dans CamNote, il s'agit d'un retour d'information à visée d'accompagnement de l'utilisateur dans le changement (Figure 1b) ;

- Dans Sedan-Bouillon, l'Extra-IHM intervient pour placer la décision de l'adaptation [Dieterich 94] dans la cour de l'utilisateur (Figure 2c). L'initiative de l'adaptation, la proposition puis son éventuelle exécution sont prises en charge par le système.

Aujourd'hui, la conception des Extra-IHM est conduite par l'application des propriétés d'utilisabilité classiques. Par exemple, en s'appuyant sur le référentiel de C. Bastien et D. Scapin [Bastien 93] :

- C'est le critère de « Guidage », sous-critère « Retour d'information » qui motive l'Extra-IHM de CamNote ;
- Celle de Sedan-Bouillon répond au critère de « Contrôle explicite ». Le retour d'information n'est donné qu'en fin d'adaptation, une fois la redistribution faite.

Demain, des critères d'ergonomie spécifiques à l'adaptation seront peut-être définis. L'*inter-utilisabilité* [Karsenty 05] va dans ce sens. Des critères existants seront peut-être affinés. C'est le cas de la *compatibilité* et de la *continuité*, aujourd'hui ré-examinées sous l'angle de la réalité mixte [Dubois 01] [Trevisan 03].

Les IHM et Extra-IHM composent l'IHM d'un système interactif plastique. Des noyaux fonctionnels métier et adaptation leur correspondent. Nous appelons *Ecosystème* le « système en contexte », c'est-à-dire le système formé des systèmes interactifs et du contexte d'usage dans lequel ils sont déployés (Figure 4). L'écosystème est, par nature, mixte. Le préfixe « Eco » fait référence à l'équilibre qu'il s'agit de préserver. Le rôle de l'adaptation est de préserver cet équilibre (c'est-à-dire les propriétés attendues) lorsque le contexte d'usage change.

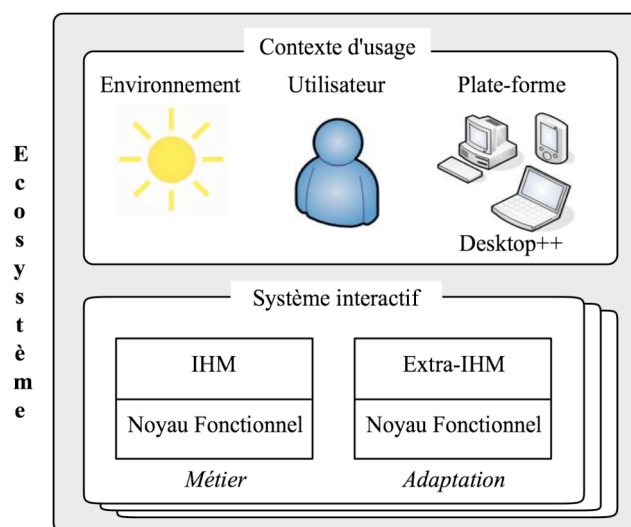


Figure 4 : Concept d'Ecosystème formé de systèmes interactifs déployés dans leur contexte d'usage.

Notre travail porte sur l'adaptation de l'IHM métier. Nous énonçons nos objectifs dans la section suivante.

2. Objectifs

Nos objectifs sont de *comprendre* la plasticité pour pouvoir *agir*, c'est-à-dire lui donner corps dans des systèmes interactifs et des outils d'ingénierie. Comprendre et agir sont des préoccupations partagées [Fekete 05]. Il s'agit ici de comprendre l'utilisateur en contexte (ses attentes fonctionnelles et extra-fonctionnelles), comprendre les variables de l'ingénierie de l'interaction Homme-Machine et cerner leurs incidences sur l'interaction pour savoir en jouer lorsque le contexte d'usage change. C'est en quelque sorte la "table de mixage" de l'ingénierie de l'Interaction Homme-Machine qu'il s'agit ici de construire.

La difficulté est l'absence d'hypothèses sur le contexte d'usage qui, de fait, place l'interaction dans un monde *ouvert*. Emettre une hypothèse définitivement éliminatrice sur le contexte d'usage, c'est prendre le risque de priver l'utilisateur d'usages séduisants. Le contexte d'usage change ainsi de statut : il n'est plus simplement l'état d'un environnement prédéfini mais participe au processus d'interaction de l'utilisateur dans un monde en perpétuelle évolution [Coutaz 05]. Il peut, en particulier, rendre possible une interaction inédite.

Notre ambition n'est pas l'universalité des systèmes interactifs [Scholtz 99]. Il ne s'agit pas de tout permettre à quiconque, en tout lieu, à tout instant. Il s'agit d'étudier et soutenir par des outils (logiciels, matériels et cadres de pensée) ce qui dans le contexte d'usage peut permettre de garantir (voire améliorer) la *valeur* de l'interaction. En IHM, la valeur était jusqu'ici classiquement exprimée en termes d'utilité et d'utilisabilité. Elle est désormais à estimer par rapport à une universalité qui tend à repousser les limites. Utilité et utilisabilité jouent un rôle de modérateurs dans cette quête de l'universel [Calvary 02].

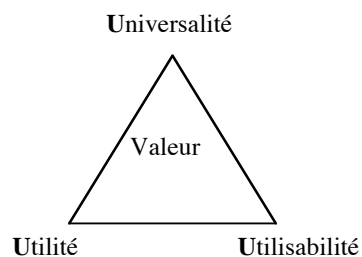


Figure 5 : L'ingénierie de l'Interaction Homme-Machine est un dosage de trois U : Utilité, Utilisabilité, Universalité.

Nous adoptons une perspective système centrée sur le système interactif. Il s'agit d'inventer, réaliser et évaluer :

- de nouvelles formes d'interaction ;
- et de nouveaux outils (logiciels, matériels et cadres de pensée) pour leur ingénierie.

La section suivante présente notre démarche pour atteindre ces objectifs.

3. Démarche

La démarche se structure en trois temps, autour d'*espaces problème et solution* :

- ❶ En premier lieu, de façon analytique, nous établissons un espace problème pour
 - a. Comprendre le problème de la plasticité et partager sa compréhension ;
 - b. Cartographier l'état de l'art et en localiser les lacunes ;
 - c. Situer une solution et en mesurer la couverture ;
- ❷ Nous définissons ensuite une approche scientifique ...
- ❸ ... à laquelle nous donnons corps le long d'axes de recherche pour construire et apprendre en retour.

Nous distinguons deux volets :

- a. La mise en œuvre de systèmes interactifs plastiques pour leur démonstration et/ou évaluation (par exemple, CamNote ou Sedan-Bouillon, démonstrateurs de remodelage, redistribution et Extra-IHM) ;
- b. L'élaboration d'outils support pour la construction de systèmes interactifs plastiques.

Nous menons nos recherches dans un esprit d'ouverture et de décloisonnement. Cet esprit est nécessaire par la nature même du domaine traité : l'ingénierie de l'Interaction Homme-Machine. Ce domaine est, en effet, à la confluence de plusieurs disciplines. Citons, à gros grain, les sciences humaines (facteurs humains, psychologie, sociologie, ergonomie) pour l'étude de l'utilisateur ; l'informatique et l'électronique pour la réalisation de systèmes et outils logiciels et matériels ; la physique et les mathématiques comme fondements ou référents. Assimiler les fondements et avancées de toutes ces disciplines est évidemment utopique. Aussi, les collaborations sont-elles précieuses. En pratique, elles nous permettent de :

- Partager le problème et confronter les expériences ;
- Compléter nos compétences.

Ce rapport s'organise selon les trois temps de la démarche :

- Le chapitre 1 pose l'espace problème ;
- Le chapitre 2 décrit l'approche scientifique ;
- Le chapitre 3 montre sa mise en œuvre au sein d'axes de recherche.

Le rapport s'ouvre sur un ensemble de perspectives montrant que le sujet est loin d'être épuisé.

Chapitre 1. Un espace problème pour comprendre



La plasticité, un sujet en trois temps.

Première mesure : établir l'espace problème pour comprendre avant d'agir.

Si l'on se risque à un parallèle avec l'Analyse des Données, nous dirions que l'espace problème est au problème ce que l'Analyse en Composantes Principales (ACP) est à l'analyse des données. Il s'agit d'identifier dans le problème les dimensions clé permettant :

- D'une part, de bien comprendre la plasticité, dans la théorie et la pratique (le concept et sa mise en œuvre) ;
- D'autre part, de créer des outils pour en faciliter l'ingénierie.

L'espace problème est donc un *modèle* du problème au sens de l'Ingénierie Dirigée par les Modèles (IDM) [Favre 04a] : il *représente* le problème pour une finalité donnée. Nous nous appuyons sur le langage et des représentations graphiques classiques telles que les espaces géométriques pour sa représentation.

Ce chapitre est structuré en deux sections. Il pose, dans un premier temps, des définitions, puis présente *Hémisphères* notre espace problème.

1. Définitions

La définition première de la plasticité axait la propriété sur la préservation de l'utilisabilité. Pour rappel : « la plasticité d'une IHM dénote sa capacité à s'adapter à sa plate-forme et son environnement d'exécution dans le respect de son utilisabilité » [Thevenin 99]. L'utilité, classiquement mise en vis-à-vis de l'utilisabilité, ainsi que la considération plus générale de l'acceptabilité du système [Nielsen 93] étaient implicites au risque en conséquence de se perdre. Par ailleurs, le degré d'universalité du système interactif n'était pas non plus mentionné, laissant supposer que, pour être dit plastique, il faille couvrir tous les contextes d'usage "possibles et inimaginables". Prenant le point de vue de l'ingénierie de l'Interaction Homme-Machine, nous posons une définition plus *opérationnelle* de la plasticité. Nous la formulons, dans un premier temps, en langage naturel puis l'adossons sur l'IDM et la théorie des graphes pour un premier pas en formalisation.

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 .

Dans cette définition, la **valeur** fait référence à un ensemble de propriétés centrées sur l'utilisateur et/ou le système. Le choix des propriétés et la définition de leurs domaines de valeurs est du ressort du concepteur. En IHM, cette notion de valeur est poussée par G. Cockton [Cockton 04] [Cockton 05]. Il estime que l'utilisabilité n'est pas suffisante : elle doit être confrontée aux véritables attentes de l'utilisateur. L'exemple donné est celui de la gestion du chauffage [Cockton 04]. Les systèmes sont peut-être utilisables au sens où ils sont faciles à utiliser et à apprendre mais aucun ne répond à la véritable motivation de l'utilisateur, à savoir économiser. Ce n'est pas par plaisir que l'usager programme son système de chauffage. C'est pour réduire sa facture. Un système de qualité devrait donc lui rendre observable le montant de son économie.

Si, en économie, la valeur est définie comme « une grandeur qui croît lorsque la satisfaction du besoin de l'utilisateur augmente ou que l'ensemble des dépenses diminue », aucun jugement de valeur n'est ici posé laissant, en conséquence, la possibilité :

- d'intégrer ou non la notion de coût ;
- de l'adosser ou non à des référentiels existants, normalisés ou non (par exemple l'ISO/IEC 9126), généralistes (pour exemples, des référentiels classiques tels que [Shackel 91] [Abowd 92] [Bastien 93] [Dix 93] [Nielsen 94] [Preece 94] [IFIP 96] [Schneiderman 97] [Constantine 99] [Van Welie 99] ou spécifiques ([Montero 04] pour le web ou [Lopez-Jaquero 04] pour l'adaptation).

Dans le choix des propriétés, les fonctions métier et adaptation sont à considérer. Par exemple, l'observabilité [IFIP 96] est un critère de valeur général, applicable aux deux parties métier et adaptation (IHM et Extra-IHM). Par opposition, la continuité de l'interaction [Trevisan 03], par exemple mesurée par le grain de reprise de la tâche, ou encore l'inter-utilisabilité [Karsenty 05] sont des critères de valeur propres à la fonction d'adaptation.

Dans la définition, le **contexte d'usage** est défini comme un triplet <Utilisateur, Plate-forme, Environnement> où :

- L'utilisateur dénote l'utilisateur du système interactif. Il peut être décrit par ses compétences métier, informatiques, etc. ainsi que par des données générales (âge, taille, etc.) ;
- La plate-forme cerne les requis matériels et logiciels nécessaires à l'interaction. Typiquement, les dispositifs d'entrée et de sortie sont à considérer ;
- L'environnement se réfère à l'espace physique hébergeant l'interaction. Il peut être décrit par ses conditions lumineuses, sonores, sociales, etc.

La définition de la plasticité donne lieu à différentes *qualifications* et *quantifications* en jouant sur ses différents mots-clé. 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 :
 - **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 **décrémentale**. S'ils varient, elle est dite **élastique**. S'ils sont stables, elle est dite **rigide**.

- l'autonomie de l'IHM dans son adaptation vis-à-vis de l'utilisateur et du système. La plasticité est dite :
 - **automatique** (versus **contrôlée**) si l'adaptation est réalisée sans le concours de l'utilisateur. Une plasticité contrôlée requiert une Extra-IHM. Une plasticité automatique ne requiert une Extra-IHM que pour l'observabilité du processus d'adaptation ;
 - **autonome** (versus **dépendante**) si l'IHM contient tous les mécanismes logiciels et ressources numériques nécessaires à l'adaptation. Si elle les contient par construction, la plasticité est dite **innée** par opposition à une plasticité **acquise**. Une plasticité dépendante est dite **plastifiabilité**. L'IHM est alors dite **plastifiable**.

Nous appelons **plastifier** l'action d'adapter une IHM à son contexte d'usage dans le respect de la valeur attendue. Nous appelons **plastification** cet acte. Selon que la plastification conserve, augmente ou réduit la valeur de l'IHM, la plastification est dite **équivalente**, **valorisante** ou **dévalorisante**.

D'un point de vue *quantitatif*, le domaine de plasticité est une mesure possible de la plasticité [Calvary 01]. Pour une valeur donnée V , nous appelons **îlot de plasticité** un ensemble de contextes d'usage C tel que l'IHM est maximale 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 6 montre quatre îlots de plasticité. Le domaine de plasticité est l'union de ces îlots.

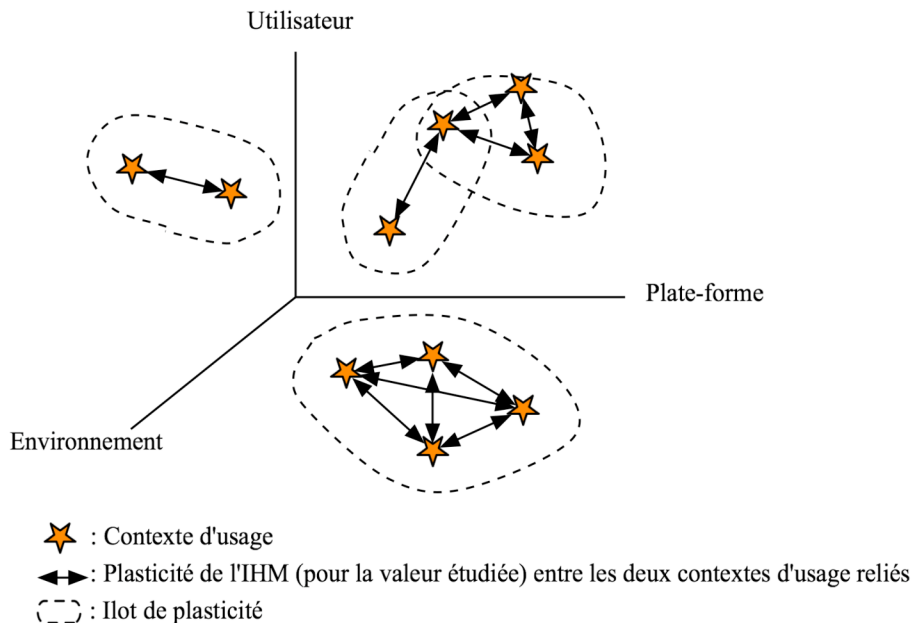


Figure 6 : Quatre îlots de plasticité.

Sur la figure 6, un contexte d'usage est représenté comme un point dans un espace à trois dimensions : l'utilisateur, la plate-forme et l'environnement. Si ce schéma permet d'illustrer la notion d'îlot de plasticité de façon informelle, il soulève des questions de fond comme :

- les contextes d'usage sont-ils des points dans cet espace ou ont-ils une "épaisseur" ?
- les contextes d'usage sont-ils ordonnés ?
- le point (0, 0, 0) a-t-il une sémantique particulière ?
- la distance entre contextes d'usage a-t-elle une sémantique ?

Se placer dans les espaces topologiques en mathématiques, et plus particulièrement dans les espaces métriques avec les notions de boule (îlot) et de voisinage est une perspective à ce travail. Pour l'instant, dans un premier pas en formalisation, nous nous appuyons sur l'IDM et la théorie des graphes.

Soit MMc un métamodèle de contexte d'usage.
 Soit G un graphe non orienté complet de sommets $C1 \dots Cn$, n contextes d'usage conformes à MMc , $n \geq 2$. Notons C cet ensemble de sommets.
 Soit V une valeur.

Une IHM I est dite **plastique** (respectivement **plastifiable**) pour V sur G si
 $\forall Ci \in C, I$ satisfait V dans Ci
 et $\forall Ci, Cj \in C / i \neq j, I$ est capable de s'adapter (respectivement d'être adaptée) au changement de contexte d'usage de Ci à Cj dans le respect de V .

Le **domaine de plasticité** d'une IHM I pour une valeur V est l'ensemble des p graphes Gi non orientés complets tels que
 $\forall i \in \{1..p\}, I$ est plastique pour V sur Gi
 et $\nexists c \notin C(Gi) / I$ est plastique pour V sur $C(Gi) \cup \{c\}$
 où $C(Gi)$ dénote l'ensemble des contextes d'usage de Gi
 et $\nexists Gk / Gk \notin \{G1 \dots Gp\}$ et I est plastique pour V sur Gk
 Les p graphes Gi sont les **îlots de plasticité** de I pour V .

Placer la définition des contextes d'usage sous l'égide de l'IDM revient à en remonter la difficulté au niveau du métamodèle. Les constituants peuvent relever de l'utilisateur, de la plate-forme et de l'environnement. Au concepteur de définir le métamodèle approprié pour son système interactif à défaut d'un métamodèle normalisé.

S'ancrer dans la théorie des graphes permet de raisonner sur les domaines de plasticité :

- Dimensionner la plasticité. La cardinalité de l'ensemble des sommets du domaine de plasticité (les contextes d'usage) peut, par exemple, servir de métrique ;
- Identifier des sommets singuliers, par exemple, sur la base de leur degré¹. Les sommets de degrés maximum (degré sortant² dans le cas d'un graphe orienté) pourraient, par exemple, constituer des contextes d'usage clé pour la conception d'IHM. Les puits³ seraient des contextes d'usage à surveiller ;
- Identifier des ruptures de plasticité par des sous-graphes non connexes⁴ dans le domaine de plasticité ;
- Définir des formes dégradées de plasticité, par exemple, sur la base de graphes orientés. La plasticité serait **directionnelle**, définie sur des chemins fondés sur les parcours⁵ possibles (fermés⁶ ou non, eulériens⁷ ou non) dans le graphe ;
- Raisonner sur la valeur via, par exemple, la valuation⁸ de graphe. A chaque chemin Cij de sommets adjacents Ci et Cj pourrait être associée la valeur garantie Vij . La valeur V de la plasticité sur le graphe G serait alors le plus grand commun diviseur

¹ Dans un graphe non orienté, le *degré* d'un sommet est le nombre d'arêtes auxquelles ce sommet appartient.

² Dans un graphe orienté, le *degré sortant* d'un sommet est le nombre d'arcs dont l'extrémité initiale est ce sommet.

³ Un *puits* est un sommet d'un graphe orienté dont le degré sortant est égal à 0.

⁴ Un graphe non orienté est *connexe* si et seulement si, pour toute paire de sommets $[a,b]$, il existe une chaîne entre les sommets a et b .

⁵ Un *parcours* de sommets dans un graphe est une liste ordonnée de sommets telle que 2 sommets consécutifs dans la liste sont adjacents dans le graphe (i.e., reliés par un arc).

⁶ Un parcours est *fermé* si le premier élément de la liste est aussi le dernier.

⁷ Un parcours est *eulérien* s'il passe une et une seule fois par chaque arête du graphe.

⁸ Une *valuation* d'un graphe est une fonction qui, à chaque arête, associe un poids (nombre réel).

des valeurs V_{ij} . Si ce plus grand commun diviseur est la valeur V annoncée, alors la plasticité est maximale en valeur.

Les définitions étant posées, la section suivante dresse l'espace problème de la plasticité.

2. Espace problème Hémisphères

En matière d'adaptation, les espaces problème sont nombreux. Ils éclairent différentes perspectives selon la préoccupation des auteurs. Nous adoptons ici le point de vue de l'ingénierie d'IHM plastiques et proposons *Hémisphères* un double espace problème faisant la séparation des préoccupations entre :

- D'une part, la constitution des directives d'adaptation (par exemple, « Si la batterie faiblit, alors migrer l'IHM sur la plate-forme la plus proche ») ;
- Et d'autre part, le cycle de vie de l'adaptation.

Les hémisphères droit (Right pour Rules) et gauche (Left pour Life cycle) sont en charge respectivement de ces préoccupations.

Hémisphères s'adresse :

- Aux chercheurs en plasticité à qui nous communiquons ainsi notre compréhension du problème ;
- Aux concepteurs de systèmes interactifs plastiques qui définiront (ou mesureront) ainsi plus facilement la couverture à donner (ou donnée) de leur IHM en matière de plasticité ;
- Aux constructeurs d'outils support qui percevront ainsi l'ensemble des leviers de la plasticité.

Les deux premières sections décrivent les hémisphères droit puis gauche. Les représentations sont planes, centrées sur le pôle de l'hémisphère. La distance d'une question au centre (le pôle) mesure le degré d'affinement de la question. La troisième section montre le caractère fédérateur de l'espace.

2.1 Hémisphère droit : Directive d'adaptation

L'hémisphère droit examine la constitution des directives d'adaptation. Une directive d'adaptation associe, sous une éventuelle *condition*, une *réaction* à un changement de contexte d'usage dans le but de préserver une certaine *valeur*. Par exemple, « Si la batterie du PC est faible et qu'un PDA s'approche, alors proposer la migration de l'IHM du PC vers le PDA » est une directive d'adaptation. Dans cet exemple, la condition porte sur l'état de charge de la batterie du PC (« Si la batterie du PC est faible »). Le changement de contexte d'usage est lié à l'arrivée du PDA (« et qu'un PDA s'approche ») : c'est un événement. La réaction est la proposition de migration (« proposer la migration de l'IHM du PC vers le PDA »). La valeur est implicite : c'est la continuité de l'interaction.

L'hémisphère droit examine les notions clé de *Condition*, *Événement*, *Action* et *Valeur* (Figure 7), c'est-à-dire les classiques règles ECA ici augmentées de la notion de *Valeur*. La figure 7 est éloquente. Elle reflète un barycentre jusqu'ici largement porté sur la partie *Action* des directives d'adaptation, c'est-à-dire sur la compréhension des leviers de la plasticité, à savoir le remodelage et la redistribution.

Nous examinons ci-après les quatre dimensions en rappelant systématiquement l'extrait de la Figure 7 qui correspond au point traité.

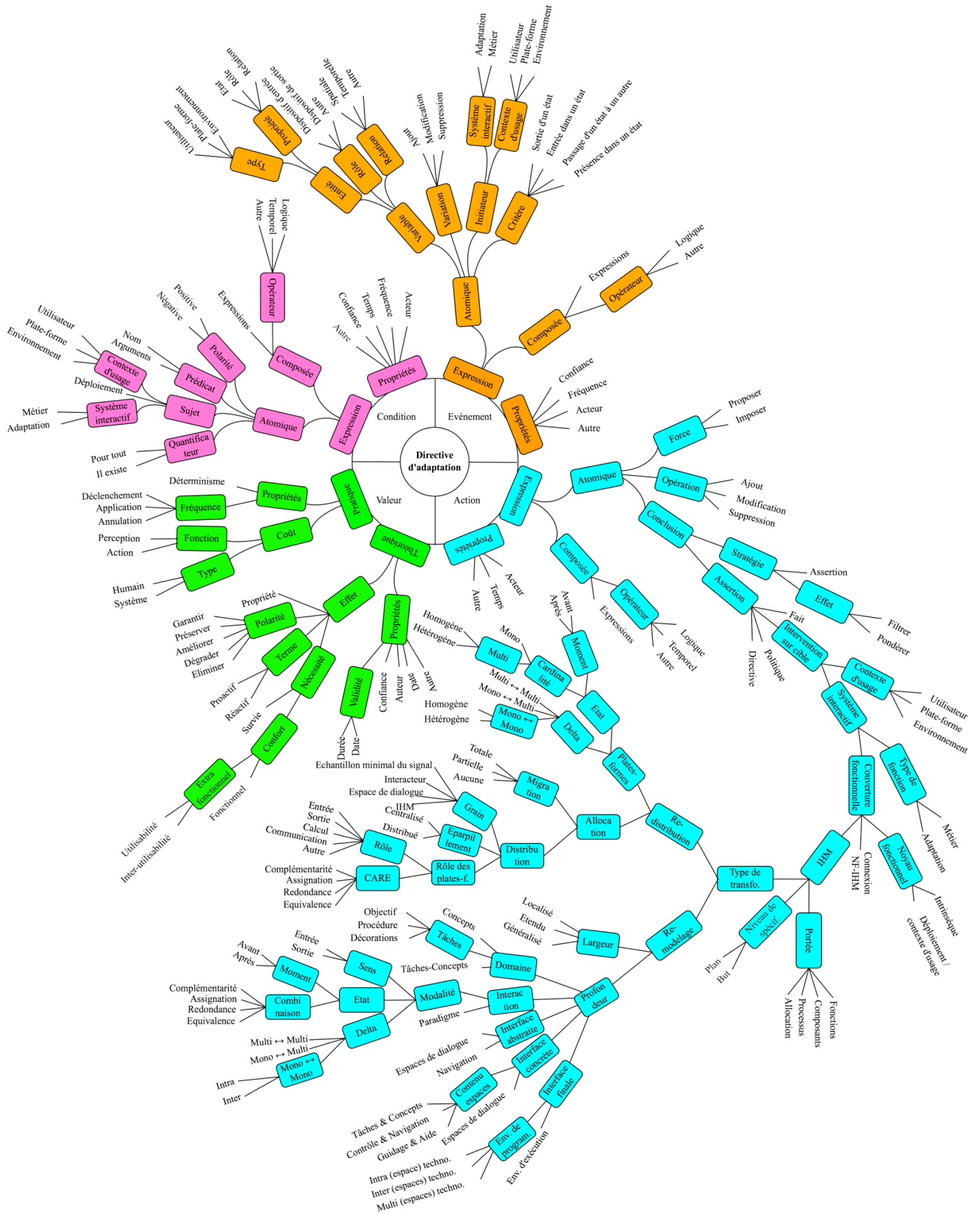


Figure 7 : L'hémisphère droit, espace problème des directives d'adaptation.

2.1.1 Condition

La *Condition* porte sur n'importe quel observable de l'écosystème. Si la condition n'est pas satisfaite, la directive ne peut pas être déclenchée. Dans son expression, la condition peut être atomique ou composée (Figure 8). Des exemples de conditions atomiques sont :

- (1) La batterie du PC est faible
- (2) Il existe un PDA à proximité de l'utilisateur
- (3) Toute plate-forme est publique
- (4) Il n'existe pas de tâche en cours
- (5) Les entrées utilisateur se font toutes sur PDA.

De façon plus générale, une condition atomique porte sur un observable (son *Sujet*) qui peut relever soit du système interactif (cas 4) dans ses parties métier ou adaptation, soit du contexte d'usage dans ses constituants <Utilisateur, Plate-forme, Environnement> (cas 1, 2 et 3), soit du déploiement du système interactif dans son contexte d'usage (cas 5). L'expression est formulée par un prédicat évalué avec d'éventuels arguments. Par exemple, pour le cas 1 : `estFaible()` ; pour le cas 2 : `estProcheDe(utilisateur)`. L'expression peut mobiliser des quantificateurs « Pour tout » ou « Il existe » (cas 2, 3, 4 et 5). Elle peut être positive (cas 1, 2, 3 et 5) ou négative (cas 4).

Une condition composée est la composition de conditions (atomiques ou composées) par le biais d'opérateurs. Par exemple :

- (1) Le PDA de l'utilisateur est allumé ET aucune tâche n'y est en cours
- (2) La batterie du PC a faibli PUIS l'utilisateur a allumé son PDA.

Ces exemples mettent en œuvre deux types d'opérateurs classiques en modélisation des tâches : les opérateurs logiques et temporels. Les opérateurs temporels vont permettre de raisonner sur l'historique de l'interaction. On pourrait imaginer d'autres types d'opérateurs.

Une condition peut être décorée de propriétés sur lesquelles il est également possible de raisonner. On peut, par exemple, imaginer une directive du genre « Si la condition *C* est fréquente alors ... ». Nous identifions comme pertinentes : l'acteur ayant évalué la condition, l'instant d'évaluation de cette condition (*Temps*), le facteur de confiance dans l'évaluation de la condition ainsi que la fréquence de satisfaction de cette condition. Ces exemples ne sont pas limitatifs.

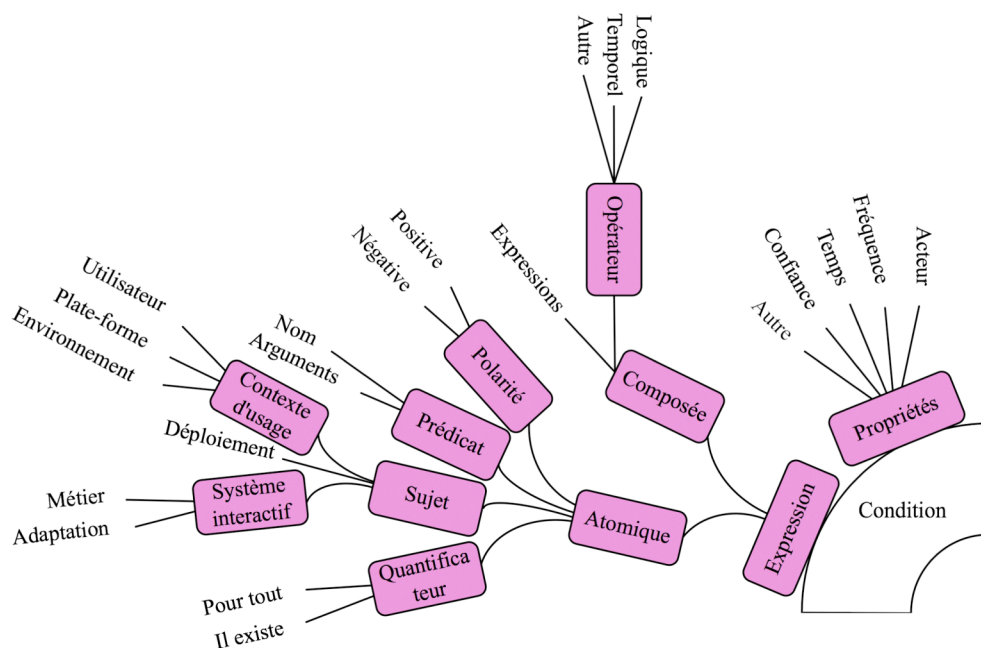


Figure 8 : Hémisphère droit – Partie Condition.

2.1.2 Événement

L'*Événement* cerne ce qui, dans le contexte d'usage, a changé et justifie une réaction. Comme éléments clé, nous identifions (Figure 9) :

- La variable du changement : le changement peut concerner l'utilisateur, la plate-forme et/ou l'environnement. Par exemple, si l'utilisateur allume son PDA, la variable est la plate-forme : elle est enrichie d'un PDA ;
- La nature de la variation : il s'agit dans ce cas de l'arrivée (*Ajout*) d'une nouvelle plate-forme ;
- L'initiateur de la variation : c'est ici l'utilisateur.

Des propriétés peuvent être posées sur l'événement comme la confiance dans la perception, l'acteur de la perception ou encore la fréquence de l'événement. Le raisonnement peut porter sur ces propriétés. On peut, par exemple, imaginer une directive du genre : « Si l'utilisateur s'assoit devant le PC et que le facteur de confiance de cet événement est de 100%, alors ... ». Ces exemples ne sont pas limitatifs.

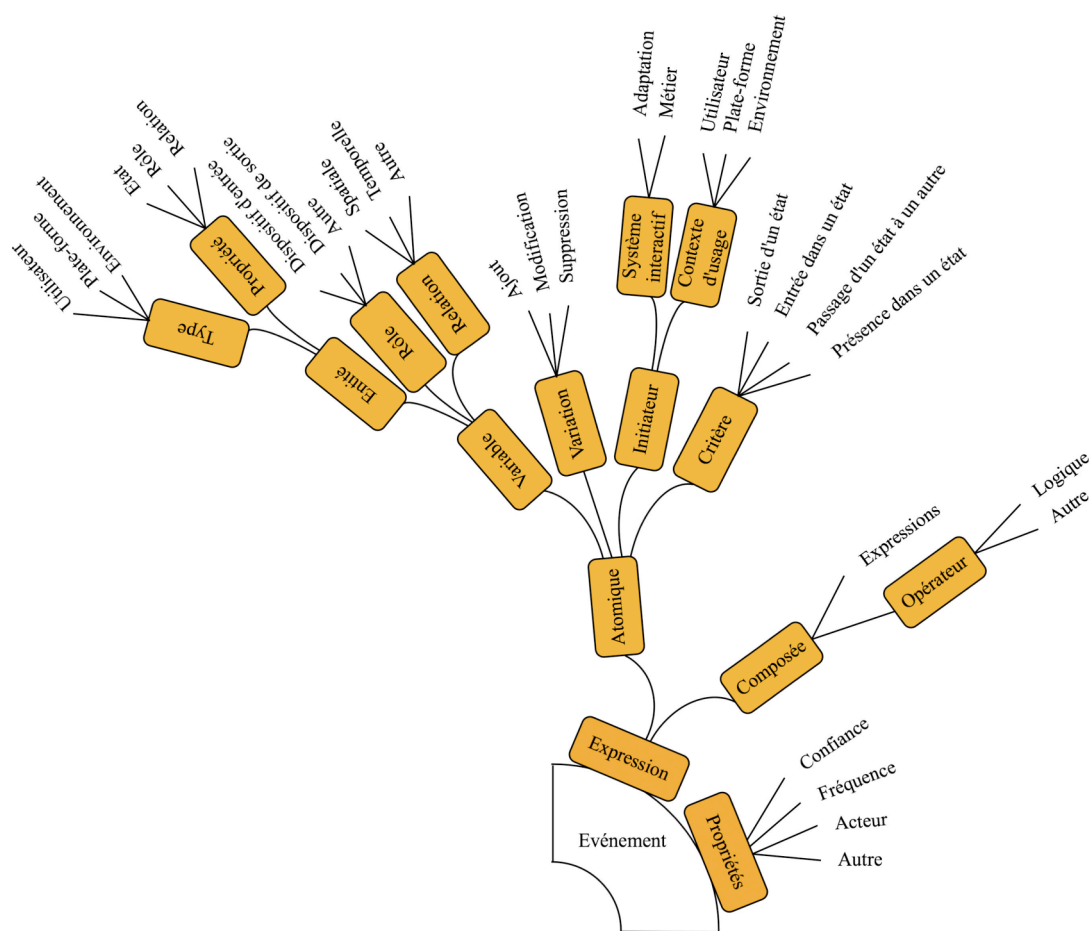


Figure 9 : Hémisphère droit – Partie Événement.

De façon plus précise, les *variables* cernent ce qui a changé dans le contexte d'usage. D'après l'ontologie de [Crowley 02], quatre types de changements peuvent survenir selon qu'ils affectent des entités, des rôles, des relations ou des associations entre entités, rôles et relations :

- Les entités relèvent du monde physique vivant ou inerte. Une entité est un regroupement d'observables. C'est à partir de la mesure d'observables que le système peut détecter la présence d'entités du monde physique, les reconnaître, les suivre, déterminer la valeur de leurs attributs et de leurs relations [Rey 05]. Nous les

généralisons ici à tous les constituants observables du contexte d'usage : utilisateur, plate-forme et environnement (*Type*) ;

- Les rôles sont les fonctions tenues par ces entités. En Interaction Homme-Machine, les rôles de dispositifs d'entrée et de sortie sont prépondérants. Jusqu'ici, ces rôles étaient assignés au traditionnel triptyque écran/clavier/souris. Mais avec l'informatique ambiante, ces objets perdent leur monopole. Ils peuvent être destitués par toute entité physique possédant les bonnes propriétés et étant observable par le système. C'est ainsi qu'une main, traditionnellement utilisée comme dispositif de pointage, devient surface d'affichage [Antoniac 02] (Figure 10). Sa proximité de l'utilisateur et la forme allongée des doigts est opportune pour un affichage tabulaire.



Figure 10 : La main, traditionnellement dispositif d'entrée, devient dispositif de sortie [Antoniac 02].

Une entité peut tenir plusieurs rôles. Inversement, un rôle peut être tenu par plusieurs entités simultanément [Rey 05] ;

- Les relations portent sur un ensemble d'entités. Elles peuvent être spatiales (« l'utilisateur est près de la plate-forme »), temporelles, fonctionnelles (« le soleil illumine la plante » [Rey 05]), etc.

Identifier les variables, c'est localiser le changement en termes d'entités (état, rôles ou relations) ainsi que de l'ensemble des rôles et relations tenus.

La *variation* caractérise le type du changement. Dans son ontologie, [Crowley 02] distingue les notions de contexte et de situation. Un contexte est défini par un ensemble de rôles et de relations, tenus par un ensemble d'entités. Un changement de contexte survient dès lors qu'un rôle ou une relation apparaît (*Ajout*) ou disparaît (*Suppression*). La situation est caractérisée par une mise en correspondance entre les entités, rôles et relations. Dès lors que ces mises en correspondance changent (*Ajout* ou *Suppression* d'associations entre les entités et les rôles/relations) ou que des entités apparaissent (*Ajout*) ou disparaissent (*Suppression*), il y a changement de situation. Nous prévoyons l'opération de *Modification* pour que des changements, par exemple, d'état d'entités puissent, si nécessaire, être intégrés à la réflexion. Cette ontologie permet d'établir un graphe de contextes et de situations. Un événement y est implicitement défini comme un arc de ce graphe. Il existe donc deux types d'événements selon qu'ils sont intra-contexte (i.e. entre situations d'un même contexte) versus inter-contextes.

[Schmidt 99] définit explicitement la notion d'événement : ce n'est pas forcément la conjonction d'une sortie de nœud (contexte ou situation) et d'une entrée dans un autre nœud. Un événement peut être, de façon plus atomique, une sortie de nœud, une entrée de nœud voire la présence dans un nœud. Nous distinguons donc les événements atomiques des événements composés. Les événements composés mobilisent des opérateurs, en particulier les opérateurs logiques (et, ou).

L'*initiateur* du changement identifie le responsable premier de la variation. Cette information peut être pertinente pour ne pas contrarier des actions utilisateur. Si, par exemple, l'utilisateur a chassé l'IHM dans une zone peu visible de l'écran, il serait inopportun dans la réaction de recentrer l'IHM. L'*initiateur* peut être le système interactif ou son contexte d'usage. Dans le système interactif, nous distinguons les parties métier et adaptation. Pour le contexte d'usage,

nous faisons référence à ses trois constituants : l'utilisateur, la plate-forme et l'environnement.

2.1.3 Action

L'action (Figure 11) spécifie la réaction à mettre en œuvre suite au changement de contexte d'usage. Par exemple :

- (1) Migrer l'IHM sur la plate-forme la plus proche
- (2) Supprimer les tâches non fréquentes
- (3) Exécuter telle IHM
- (4) Privilégier les remodelages
- (5) Eliminer les redistributions automatiques
- (6) Déclarer l'utilisateur fatigué
- (7) Ajouter telle directive
- (8) Exécuter la première directive applicable.

Dans ces exemples, les actions sont toutes atomiques. On peut toutefois imaginer des actions composées combinant des actions (atomiques ou composées) par le biais d'opérateurs (par exemple, logiques ou temporels). Les actions, qu'elles soient atomiques ou composées, peuvent être décorées de propriétés. Nous identifions comme pertinentes la spécification de l'acteur en charge de leur exécution ainsi que l'instant auquel elles doivent être exécutées (*Temps*). Ces exemples ne sont pas limitatifs.

Une action atomique est émise avec une certaine force : l'action est proposée ou imposée. Les actions sont régies par des politiques de contrôle, par exemple « Exécuter toute action imposée ». Le cas 8 est un exemple de politique de contrôle ajoutée à l'exécution. L'ajout, la suppression ou la modification de politiques de contrôle sont un type d'action. Ce sont des actions de type *Assertion* par opposition aux *Stratégies*.

Les stratégies manipulent des assertions : elles les filtrent (cas 5) ou les pondèrent (cas 4). On distingue quatre types d'assertions : les politiques (cas 8) ; les faits (cas 6) ; les directives d'adaptation (cas 7) et les interventions sur cible (cas 1, 2 et 3). Ce sont les interventions sur cible qui ont jusqu'ici été le plus étudiées. Elles agissent sur le contexte d'usage (par exemple, allumer la lumière) ou le système interactif. Dans le cas du système interactif, on affine fonctionnellement la cible selon, d'une part, les parties métier ou adaptation concernées et, d'autre part, d'un point de vue de l'architecture logicielle, les composants (noyau fonctionnel ou IHM) et connecteurs (noyau fonctionnel - IHM) concernés. Nous développons ici les interventions sur l'IHM. Pour le noyau fonctionnel, nous nuancions simplement selon que l'adaptation concerne le noyau fonctionnel intrinsèquement ou son déploiement sur le contexte d'usage.

Les interventions sur l'IHM peuvent être étudiées selon la perspective de l'architecture logicielle (*Portée*) : quels sont les éléments (fonctions, composants, processus) et allocations entre éléments (fonctions aux composants, composants aux processus, processus aux ressources physiques) concernés ? Ce point de vue met à profit les acquis en architecture logicielle [Coutaz 01] sans pour autant préjuger d'un quelconque style d'architecture (ARCH par exemple [Arch 92]). Les éléments et allocations concernés dépendent du type de l'adaptation. On distingue deux leviers de plasticité : le remodelage et la redistribution. Le remodelage (cas 2) agit à état de distribution constant de l'IHM sur les ressources d'interaction, contrairement à la redistribution (cas 1). Remodelage et redistribution peuvent être spécifiés en termes de but ou de plan. Le but fixe l'objectif à atteindre (par exemple, migrer) sans imposer de solution conceptuelle et/ou implémentationnelle (cas 1 et 2), contrairement au plan (cas 3) dans lequel tous les degrés de liberté sont fixés.

l'itération, l'optionalité, etc. La figure 12 montre un exemple de modèle de tâches pour une application de domotique. L'utilisateur contrôle sa température à domicile (*Manage home temperature*) en traitant, de façon itérative (décoration *), les différentes pièces de la maison. Traiter une pièce, c'est spécifier une commande (*Specify command*) puis éventuellement consulter le retour d'information donné (*Check feedback* avec sa décoration d'optionalité). Spécifier une commande, c'est spécifier la pièce d'intérêt (*Specify room*) et l'action à lui appliquer : soit contrôler sa température (*Check temperature*), soit modifier sa température (*Set temperature*).

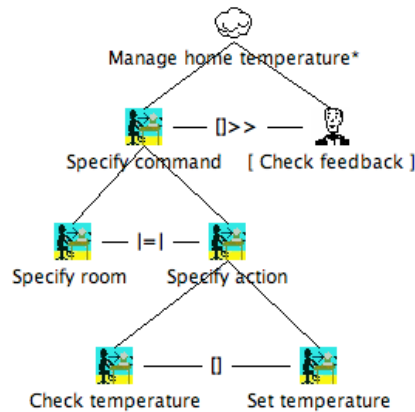


Figure 12 : Exemple de modèle de tâches CTT édité avec l'environnement CTTE.

Un modèle des tâches est, en théorie, établi par un spécialiste en facteurs humains. Il ne devrait présager d'aucune modalité. Ainsi, l'ordre laissé indifférent dans le modèle entre les tâches *Specify room* et *Specify action* pourrait, en pratique, devenir une séquence. En graphique, on imagine volontiers une carte de la maison sur laquelle l'utilisateur cliquerait dans une pièce d'intérêt pour percevoir, et éventuellement, modifier la température de cette pièce (Figure 13a) : l'ordre serait alors *Specify room* puis *Specify action*. En vocal, on pourrait, par contre, très bien imaginer que l'utilisateur dise « Quelle est la température du salon ? ». La séquence est alors en ordre inverse : *Specify action* puis *Specify room*. Les degrés de liberté laissés au niveau du modèle des tâches sont fixés au fil de la conception. En pratique, bien souvent, les modèles de tâches sont déjà dépendants de la modalité. La figure 13b est un modèle de tâches orienté graphique pour l'application de domotique.

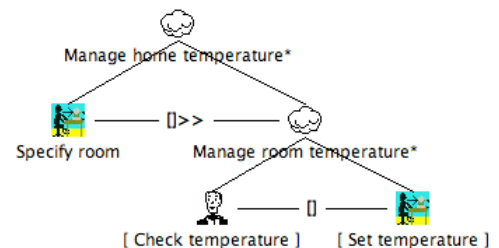


Figure 13 : Maquette graphique (a) compatible du modèle des tâches des figures 12 et 13b.

- L'interface abstraite structure l'IHM en espaces de dialogue (aussi dits espaces de travail ou unités de présentation) et fixe la navigation entre ces espaces. Un espace de dialogue est un « lieu d'activité virtuel offrant les éléments nécessaires à la réalisation d'une ou plusieurs tâches » [Normand 92]. Clairement, l'identification des espaces de dialogue répond à une préoccupation de groupement (spatial dans le cas d'IHM graphiques). Cette préoccupation est motivée par le critère de « Guidage-Groupement/Distinction entre items » tel que formulé dans le référentiel de [Bastien 93]. Depuis, une réflexion au-delà du graphique est menée. Les espaces de dialogue pourraient aussi avoir la vocation de résolution de références ou d'erreurs de dialogue en vocal. Les propriétés CARE (Complémentarité Assignment Redondance Equivalence) [Coutaz 95] pourraient intervenir à ce niveau pour exprimer des besoins de redondance dans un contexte d'usage bruité par exemple.

Nous distinguons trois types d'espaces de dialogue : les *espaces de production* permettant la réalisation des tâches élémentaires (les feuilles du modèle de tâches) dans leur spécification et contrôle avec le noyau fonctionnel ; les *espaces de navigation* donnant accès, pour une tâche donnée, à ses sous-tâches ; et enfin les *espaces de guidage* dédiés à l'accompagnement de l'utilisateur dans l'interaction en incitation et en retour d'information [Bastien 93]. La zone basse de l'écran est typiquement un espace de guidage approprié pour l'affichage de messages à lecture optionnelle [Nogier 05]. La figure 14b montre une maquette d'IHM structurée en deux espaces de navigation (les bandeaux haut et gauche faits respectivement d'onglets et de liens hypertexte) ainsi que d'un espace de production (la zone centrale). Cette maquette est conforme au modèle des tâches de la Figure 14a (critère de « Compatibilité par rapport à la tâche » [Bastien 93]). Les onglets (respectivement liens hypertexte) sont la représentation des "ou" non exclusifs entre les sous-tâches T_i (respectivement T_{1j}) affinant la tâche T (respectivement T_1). L'organisation spatiale des espaces est du ressort de l'interface concrète. Elle est ici conforme au tracé régulateur des ergonomes [Nogier 05] (critère de « Compatibilité par rapport aux autres environnements » [Bastien 93]).

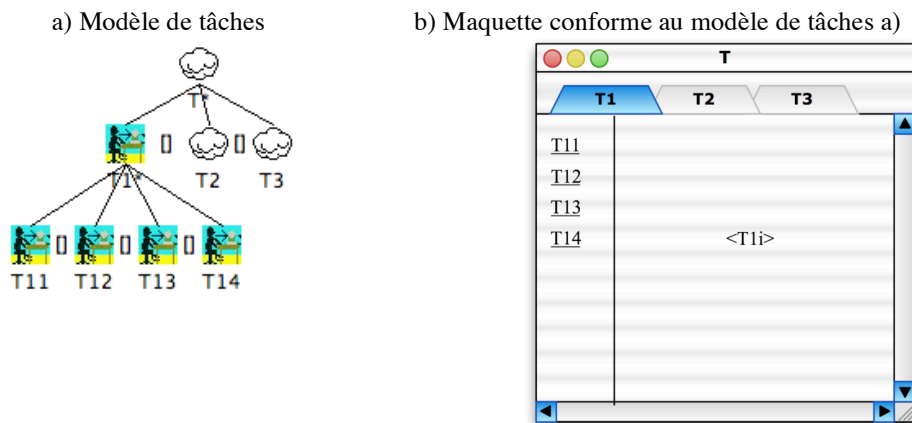


Figure 14 : Illustration de deux types d'espaces de dialogue : espaces de navigation (bandeaux haut et gauche) et de production (zone centrale).

- L'interface concrète fait le choix d'interacteurs pour :
 - Les espaces de dialogue alors réifiés en fenêtres ou canevas dans le cas d'IHM graphiques ;
 - Le contenu des espaces de dialogue : pour les espaces de production, les tâches élémentaires et concepts du domaine sont classiquement spécifiés par des boutons radio, cases à cocher, texte, etc. Le contrôle est typiquement assuré par les boutons « Valider » et « Annuler ». La navigation entre espaces

est classiquement incarnée par des séparateurs (espace ou trait) ou des objets de navigation (boutons, liens hypertexte, etc.). Le guidage est souvent textuel ;

- L'interface finale fait le choix d'un environnement de programmation et d'exécution.

Ces quatre niveaux d'abstraction sont généraux. Aussi ont-ils été repris en plasticité. En particulier, le projet européen CAMELEON en fait sa colonne vertébrale [Calvary 03]. L'adaptation peut s'ancrer à tout niveau d'abstraction :

- Tâches-Concepts par la suppression d'une tâche qui, par exemple, n'aurait pas de sens sur la plate-forme ciblée ;
- Interface abstraite par une réorganisation des espaces de dialogue suite, par exemple, à un changement de modalité ;
- Interface concrète par la suppression, par exemple, de la manipulation directe du fait de l'utilisation d'une boule roulante (changement de paradigme d'interaction) ;
- Interface finale par le choix d'une version HTML plutôt que Swing pour une consultation par Internet.

[Florins 06] reprend ces quatre niveaux d'abstraction pour classer les règles d'adaptation. En réalité, le niveau d'abstraction doit être vu comme un point d'ancrage de l'adaptation dans la colonne vertébrale. En effet, l'impact d'une règle est longitudinal. Typiquement, la suppression d'une tâche peut, en ricochet, supprimer un espace de dialogue dans l'interface abstraite et consécutivement tous les interacteurs correspondants dans les interfaces concrète et finale. Si intuitivement, il semble préférable d'ancrer l'adaptation à faible niveau d'abstraction (interface finale de préférence), cette intuition est, en réalité, à modérer selon les décorations de tâches. Typiquement, si une tâche n'est pas fréquente, peut-être est-il préférable de supprimer cette tâche plutôt que d'affecter la présentation concrète d'une tâche fréquente. Des études sont à mener sur ce point.

L'hémisphère droit (Figure 11) inclut cette préoccupation du niveau d'abstraction dans la *profondeur* du remodelage. Il précise les niveaux en distinguant pour :

- Le domaine : la description des tâches en termes d'objectif (par exemple, *Manage home temperature*), de procédure (la décomposition en sous-tâches) et de décorations (par exemple, la fréquence et l'optionnalité) ; la description des concepts et du lien entre tâches et concepts (*Tâches-Concepts*). Ce lien se fait classiquement en décorant les tâches des concepts qu'elles manipulent. Cette décoration est d'importance : les concepts sont les entités représentées dans les espaces de dialogue, ni plus ni moins au risque sinon de transgresser soit le critère de « Charge de travail » (ni plus) soit le critère de « Compatibilité par rapport à la tâche » (ni moins) [Bastien 93] ;
- L'interface abstraite : l'identification des espaces de dialogue (production, navigation et guidage) et la spécification de la navigation ;
- L'interface concrète : la présentation des espaces de dialogue (fenêtres ou canevas en graphique) et du contenu de ces espaces (tâches et concepts ; contrôle et navigation ; guidage et aide [Vanderdonckt 05]). On peut en effet imaginer augmenter le niveau de guidage si l'utilisateur peine ou commuter sur une aide plus étoffée ;
- L'interface finale : l'environnement de programmation et d'exécution. Pour l'environnement de programmation, on peut préciser si le changement se fait intra technologie (ou espace technologique) (par exemple, en restant en XML) ; ou entre (*Inter*) technologies (ou espaces technologiques) (par exemple, de Swing à HTML) ou en combinant (*Multi*) différentes technologies (ou espaces technologiques). Par exemple, CamNote panache (*Multi*) HTML/PHP et C++ : la télécommande est en HTML/PHP tandis que les diapositives sont en C++.

Nous enrichissons ces quatre points d'une préoccupation de l'ordre de l'*Interaction*. L'adaptation peut consister en un changement de paradigme d'interaction (par exemple, la

manipulation directe) ou de modalité humaine (visuelle, sonore, auditive, etc.). Pour la modalité, nous distinguons son *sens* (entrée/sortie). Nous raisonnons ensuite en termes d'*Etat* ou de changement d'état (*Delta*). L'état peut s'étudier en termes des propriétés CARE (*Combinaison*) analysées avant et après remodelage. Le changement d'état distingue les remodelages selon que l'IHM était, reste ou devient *Mono* versus *Multimodale*. Si l'IHM reste monomodale, la modalité peut toutefois changer. On distingue les remodelages intramodaux (i.e., qui conservent la modalité) des remodelages intermodaux (i.e. qui changent de modalité).

La deuxième dimension du remodelage est sa *largeur*. Il peut être intéressant de mesurer l'étendue d'une adaptation. Nous proposons trois niveaux : localisé, étendu et généralisé. Si au niveau Tâches-Concepts, une modification peut être localisée (suppression d'une tâche, par exemple), les effets sont certainement plus généralisés au niveau de l'interface concrète. En effet, le critère de « Homogénéité-Cohérence » [Bastien 93] milite pour une propagation des modifications à tous les homologues de l'élément concerné. Les changements de fonte, de couleur, etc. (critère de « Signifiante des codes et dénominations » [Bastien 93]) auront typiquement très probablement un effet étendu voire généralisé.

La *redistribution* a été intégrée plus tard à la réflexion [Balme 04], comprenant que, d'une part, les IHM ne sont plus forcément centralisées et que, d'autre part, elles peuvent migrer dynamiquement, changeant alors, à l'exécution, d'état de distribution. Dès lors, de nouveaux concepts ou propriétés fleurissent. On parle :

- d'IHM migrables [Grolaux 04],
- d'IHM détachables [Grolaux 05],
- ou encore de déplier d'IHM pour déplier, sur grande surface, une IHM confinée dans un petit appareil tel qu'un téléphone. Cette fonction de dépliement est présente dans « Surface » de Microsoft : lorsque l'appareil photo est déposé sur la table, la photo tout juste prise s'affiche sur la table permettant alors à l'utilisateur de la manipuler (Figure 15).



Figure 15 : Surface de Microsoft.

Nous caractérisons la redistribution par deux éléments : d'une part, les plates-formes utilisées, d'autre part, la distribution (*Allocation*) de l'IHM sur ces plates-formes. Concernant les plates-formes, il est possible de raisonner en terme d'*état* ou de changement d'état (*delta*). L'état peut s'analyser avant et après redistribution (*Moment*) en raisonnant sur la cardinalité des plates-formes : *Mono* versus *Multi*. Dans le cas du multi-plates-formes, il peut être pertinent de distinguer les clusters (i.e. ensembles de plates-formes) homogènes (i.e. ne contenant que des plates-formes de même gamme, par exemple des PC) versus hétérogènes (composés de plates-formes diverses, par exemple des PC et PDA). Le changement peut raisonner sur le passage entre Mono et Multi-plates-formes. Si le cluster reste élémentaire (*Mono*), il peut toutefois y avoir changement de plate-forme. Il peut alors être pertinent d'identifier si ces plates-formes sont de même gamme (homogène) ou non (hétérogène).

Concernant l'allocation, on peut de même raisonner en terme d'état de distribution ou de changement d'état. Le changement d'état peut s'opérer par migration. La migration peut être totale (l'IHM migre dans son intégralité) ou partielle (seules des parties de l'IHM migrent). Mais l'état de distribution peut changer autrement que par migration : afficher une information de façon redondante sur une autre plate-forme en est un exemple. Les propriétés CARE [Coutaz 95] sont utiles pour raisonner sur le rôle des plates-formes. Dans « Surface », la table est *assignée* à la manipulation de l'image. Dans Sedan-Bouillon, le PDA et le PC sont *complémentaires*, permettant ensemble à l'usager de parcourir des sites Web. On pourrait imaginer un affichage redondant sur le PC et le PDA offrant ainsi à l'utilisateur une *équivalence* pour la réalisation de sa tâche. On pourrait aussi, par sécurité, demander à l'utilisateur de décliner son identité à la fois sur PC et PDA. Ceci constituerait une *redondance* en terme des propriétés CARE. Cet usage des propriétés CARE est intéressant pour explorer les possibilités en matière de distribution. A noter que ces propriétés, originellement introduites pour raisonner sur la modalité ont déjà, à plusieurs reprises, été détournées de leur sujet d'origine pour raisonner sur d'autres préoccupations, comme les collecticiels par exemple [Salber 95]. Nous pouvons ici les utiliser pour raisonner sur les rôles d'entrée, de sortie, de calcul, de communication ou toute autre fonction des plates-formes.

D'un point de vue système, on dira une IHM *distribuée* (versus *centralisée*) dès lors qu'elle est rendue sur au moins deux plates-formes, quels que soient les rôles tenus par ces plates-formes. Ainsi, même si par les prouesses technologiques (amincissement des bords des écrans), l'utilisateur a l'illusion de ne voir qu'une surface face à deux écrans juxtaposés, l'IHM sera dite distribuée parce qu'affichée sur deux écrans gérés par des CPU différentes. Aujourd'hui, l'épaisseur des bords est telle que l'utilisateur n'a pas encore cette illusion (Figure 16). Cette limite physique constitue un frein à l'étalement des IHM au grain du pixel.



Figure 16 : Le bord des écrans nuit à une distribution d'IHM au grain du pixel.

Nous identifions quatre *grains* de distribution :

- l'échantillon minimal du signal : le pixel en graphique (Figure 16), la fréquence en vocal ;
- l'interacteur : en graphique, un libellé ou une image qui, dans ce cas, ne seront pas sectionnés ;
- l'espace de dialogue : en graphique, des fenêtres qui ne seront pas morcelées. C'est le cas dans Sedan-Bouillon ;
- l'IHM complète.

Le choix dépend, en partie, de l'approche scientifique et technologique choisie.

2.1.4 Valeur

En économie, la valeur mesure le ratio entre un bénéfice et un coût. Dans cette lignée mais sans jugement de valeur, nous permettons la prise en considération de ces deux éléments (Figure 17) : le bénéfice d'application, d'une part, que nous généralisons ici en *effet* de façon à ne pas se limiter à des effets positifs et le *coût* de mise en œuvre, d'autre part, lié à

l'exécution de la directive. Le coût est d'ordre pratique par opposition à l'effet qui lui est théorique.

L'effet cerne les retombées attendues de la directive. Il se formule en termes de propriétés exprimées dans un référentiel donné (par exemple, [Bastien 93]). Ces propriétés peuvent être :

- garanties par l'application de la directive : elles n'étaient pas assurées avant l'application ; elle le deviennent. L'observabilité d'un concept [IFIP 96] pourrait en être un exemple ;
- préservées : elles étaient satisfaites ; elle le restent ;
- améliorées : elles étaient partiellement satisfaites ; elles le sont un peu plus. Par exemple, la charge de travail en terme de densité informationnelle [Bastien 93] peut s'améliorer par une migration sur grand écran ;
- dégradées : elles étaient partiellement satisfaites ; elles le sont un peu moins. Par exemple, la migration sur petit écran peut augmenter les tâches articulatoires de navigation et, en conséquence, défavoriser la charge de travail en terme d'actions physiques [Bastien 93] ;
- ou éliminées : elles étaient satisfaites ; elles ne le sont plus. Un exemple pourrait être la suppression du guidage utilisateur par manque de surface d'affichage.

Si l'analyse en termes de propriétés est délicate, l'effet peut être exprimé, de façon plus globale, par la *nécessité* ou non de l'application de la directive : la directive relève-t-elle de la survie du système interactif ou du confort de l'utilisateur ? S'il s'agit du confort de l'utilisateur, est-il d'ordre fonctionnel, c'est-à-dire relatif à l'utilité du système interactif, ou extra-fonctionnel lié à son utilisabilité (IHM) ou inter-utilisabilité (Extra-IHM) ?

Que la directive relève de la survie du système interactif ou du confort de l'utilisateur, elle peut être appliquée de façon proactive, c'est-à-dire anticipatoire, ou réactive pour faire face à une réalité. Nous appelons *terme* cette dimension de l'effet.

La valeur théorique peut être décorée de propriétés, utiles au raisonnement. Nous identifions comme pertinentes la validité de la directive (date ou durée de péremption), la confiance accordée à la directive, son auteur et sa date d'édition. Ces exemples ne sont pas limitatifs.

La valeur pratique prend en considération le coût d'application de la directive. Ce coût se mesure d'un point de vue système (ressources numériques et physiques nécessaires au calcul, à la communication et à l'interaction) mais aussi humain (la charge perceptuelle, cognitive et motrice typiquement) pour la perception (condition et événement) et l'action. L'estimation du coût est un point dur en plasticité.

La valeur pratique peut aussi être décorée de propriétés. Ces propriétés sont évaluées par la pratique. Nous identifions comme pertinentes la fréquence de déclenchement de la directive, sa fréquence d'application et d'annulation. Nous ajoutons aussi son déterminisme : ce déterminisme intègre, en particulier, le niveau d'abstraction de la directive.

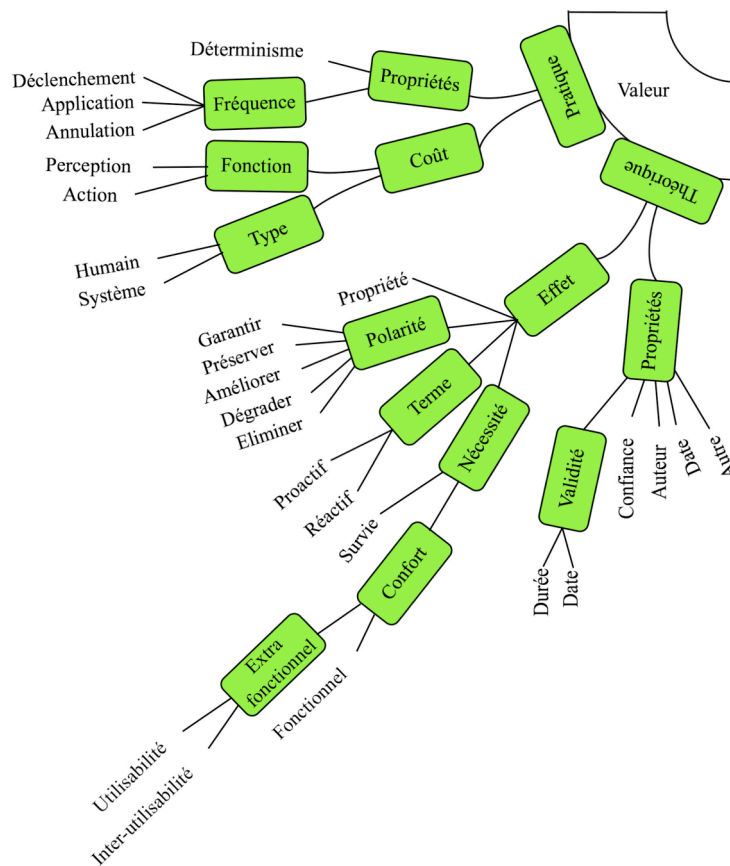


Figure 17 : Hémisphère droit – Partie Valeur

Les directives d'adaptation étant définies, la section suivante étudie leur cycle de vie.

2.2 Hémisphère gauche : Cycle de vie

L'hémisphère gauche identifie quatre temps dans le cycle de vie de l'adaptation : la définition des directives d'adaptation, leur exécution, leur évaluation et enfin la capitalisation de l'expérience. Pour chaque temps, nous soulevons les questions de *qui* fait *quoi*, *où*, *quand* et *comment* (Figure 18). Le *pourquoi* n'est pas considéré, relevant, de façon générale, d'une justification de la conception. Les travaux tels que QOC (Question Options Critères) [MacLean 96] entrent dans ce cadre.

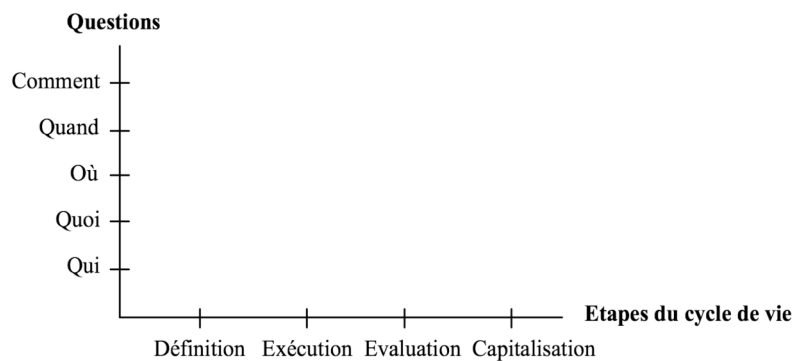


Figure 18 : Deux dimensions orthogonales, fondatrices de l'hémisphère gauche.

Nous adoptons une représentation hémisphérique plane qui ventile les cinq questions par étape du cycle de vie (Figure 19). Chaque question est affinée en mettant à profit les espaces

problèmes existants ainsi que les compréhensions récentes. On note un effort largement porté sur la dimension *Comment*.

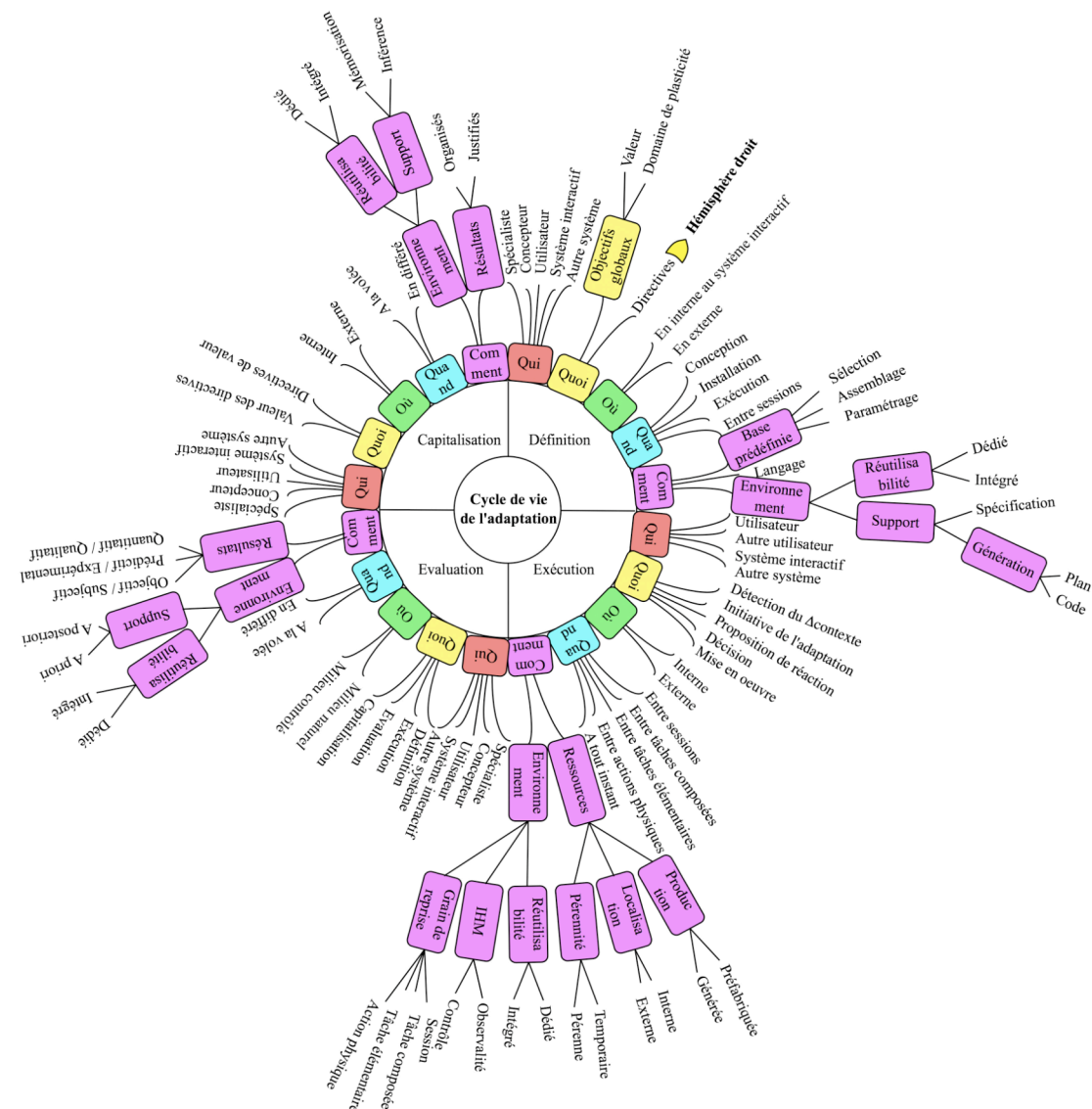


Figure 19 : L'hémisphère gauche, espace problème du cycle de vie de l'adaptation.

L'ambition est de fédérer et compléter les taxonomies existantes qui, en général, se concentrent sur une question particulière. Par exemple, la taxonomie de [Totterdell 90] développe les questions du *Qui* et du *Quoi* à gros grain. Pour rappel, cette taxonomie identifie six niveaux croissants de sophistication en matière d'adaptation (extrait de [Thevenin 01]) :

- Système câblé (designed system) : le comportement du système est défini à la conception. Aucune adaptation n'est possible.
- Système adaptable (adaptable system) : le système est personnalisable sur intervention explicite de l'utilisateur qui peut agir sur les paramètres fixés par le concepteur.
- Système adaptatif (adaptive system) : le système est doué de discrimination. Il sait reconnaître la situation (parmi plusieurs déclencheurs fixés par le concepteur) et adopte la réaction (recommandation) elle aussi fixée par le concepteur pour cette situation. Mais le système est incapable d'évaluer l'effet de sa réaction.
- Système autorégulateur (self-regulating adaptive system) : le système est doué de facultés d'auto-évaluation. Non seulement il reconnaît la situation parmi les déclencheurs prévus par le concepteur, mais utilise, pour effectuer son choix de

réaction, une fonction de feedback sur lui-même par essai-erreur. Autrement dit, il utilise une métrique qui évalue l'effet de la réaction retenue précédemment pour ce même déclencheur.

- Système automédiateur (self-mediating system) : le système a les caractéristiques de l'autorégulation, mais en plus, est capable de résoudre le problème de l'adaptation par planification et peut évaluer a priori l'effet d'une réaction (alors qu'un système autorégulateur n'évalue l'effet qu'a posteriori).
- Système automodificateur (self-modifying adaptive system) : c'est la phase ultime du système adaptatif autonome. Il s'agit d'un système doué de généralisation qui, par ses méta-connaissances, a la capacité d'apprendre de nouveaux déclencheurs et de nouvelles réactions.

Tous les travaux menés sur les propriétés *Self** (*Auto**), en particulier en Autonomic Computing [Kephart 03], relèvent de cette même préoccupation : affiner le *Quoi* et classer les systèmes interactifs selon que les fonctions correspondantes sont prises en charge par le système interactif lui-même ou un tiers. Mais la suprématie n'est pas forcément au « tout Auto ». En effet, au-delà du contrôle que l'utilisateur apprécie de garder, des études citées dans [Stuerzlinger 06] montrent que l'utilisateur n'est pas moins efficace avec des systèmes adaptables que des systèmes câblés ou adaptatifs dès lors qu'il est conscient des possibilités d'adaptabilité et que l'IHM reste simple. L'étude est menée dans le cas particulier des menus [Findlater 04], sans avoir toutefois été généralisée.

Les classifications à la [Totterdell 90] occultent des questions telles que le *Où*, *Quand*, *Comment*, en revanche présentes dans d'autres espaces [Thevenin 01] [Calvary 02] [Vanderdonckt 05] [Calvary 06]. L'hémisphère gauche rassemble les acquis pour aider le concepteur à se poser les bonnes questions.

2.2.1 Définition

Par “définition”, nous entendons la spécification, éventuellement par génération, des consignes d'adaptation. Les autres temps du processus de développement (conception, codage) ne sont pas considérés, les variations en termes d'acteur en particulier étant limitées. Dans les spécifications (*Quoi*), nous distinguons la spécification des objectifs globaux de l'adaptation, de la spécification des directives d'adaptation (Figure 20). Par objectifs globaux, nous entendons la spécification, d'une part, de la valeur à garantir, d'autre part, du domaine de plasticité à assurer pour cette valeur. Les directives ne sont pas affinées : l'hémisphère droit y est consacré.

Les spécifications peuvent être faites par (*Qui*) des spécialistes en facteurs humains ou en adaptation, le concepteur, l'utilisateur final, le système interactif ou un autre système (applicatif ou infrastructure) pour peu que ces derniers soient dotés de mécanismes d'inférence (réseaux bayésiens par exemple). Dans CamNote et Sedan-Bouillon, les objectifs globaux de l'adaptation ne sont pas explicites. Les directives sont spécifiées par le concepteur. L'outil de spécification est le langage de programmation du système interactif.

Les outils de spécification (*Comment*) dépendent, bien entendu, de l'acteur en charge de la spécification. On peut imaginer différents outils allant d'une base prédéfinie d'objectifs globaux et/ou directives [Florins 06] à des environnements de spécification, en passant par des langages de spécification [Ganneau 07]. Dans le cas de bases prédéfinies, l'acteur procède par sélection/désélection des éléments prédéfinis. Selon le grain de ces éléments, on peut imaginer une approche par assemblage d'éléments plus atomiques. On peut aussi imaginer un paramétrage de ces éléments. En matière d'environnements de spécification, nous distinguons les environnements selon qu'ils permettent ou non la génération de plans à partir d'objectifs [Clerc 03] et/ou la génération de code (par exemple, l'instrumentation automatique du

système interactif pour que les directives soient appliquées sur changement de contexte d'usage). Ces environnements peuvent être génériques, dédiés à l'adaptation, et donc réutilisables ou, au contraire, intégrés au système interactif, compromettant, en conséquence, leur réutilisation. Tous ces outils prennent une facture particulière dès lors que l'utilisateur est l'utilisateur final du système interactif : c'est le champ du *end-user programming*.

La programmation par l'utilisateur final non-informaticien est un sujet de recherche initié il y a une trentaine d'années [Smith 77]. Ces travaux ont surtout cherché à définir des notations simplifiées : langages de script textuels (HyperTalk) ou graphiques (Visual AgenTalk) [Repenning 06], programmation par démonstration et par l'exemple [Cypher 93], construction de macros (à-la Emacs). Ces techniques ont été appliquées essentiellement à des domaines ciblés comme la CAO [Girard 92], les tableurs [Burnett 03], la bio-informatique [Letondal 05] ou, tels Alice [Conway 97] et HANDS [Pane 02], à l'apprentissage de la programmation. Depuis peu, la recherche s'ouvre sur le Web avec la possibilité, pour l'utilisateur, de construire des macros : avec Koala, un utilisateur enregistre ses actions sur une page web (par exemple, la saisie des champs d'un formulaire), il peut les rejouer et les publier sur un wiki pour les partager avec les membres de sa communauté (qui peuvent réutiliser le formulaire prérempli) [Little 07].

Les travaux sur la programmation des espaces ambiants par l'utilisateur final sont encore très prospectifs. Dans Speakeasy, l'utilisateur peut construire des requêtes de recherche de ressources d'interaction avec filtrage [Newman 02]. Avec Jigsaw, l'utilisateur, au moyen d'un éditeur graphique dédié, construit des programmes simples par assemblage de pièces de puzzle du genre « si quelqu'un sonne à la porte, prendre une photo et la transférer sur le PDA » [Rodden 04]. Les media cubes [Blackwell 01], ICAP [Sohn 06] et a CAPella [Dey 04] s'attaquent à des problèmes similaires, mais, comme les autres, n'incluent pas d'outils de mise au point façon Whyline [Myers 06].

Selon les outils (*Comment*), l'acteur (*Qui*), etc., la spécification peut être faite à différents moments (*Quand*). On distingue quatre moments clé : la conception, l'installation du système interactif, son exécution et l'entre sessions, requérant alors l'arrêt du système interactif puis son redémarrage. Dans CamNote et Sedan-Bouillon, la spécification est faite à la conception.

Les objectifs et/ou directives peuvent être mémorisés (*Où*) dans le système interactif lui-même (interne) ou de façon externe dans des fichiers de configuration par exemple. L'appellation interne/externe est reprise de [Oreizy 99].

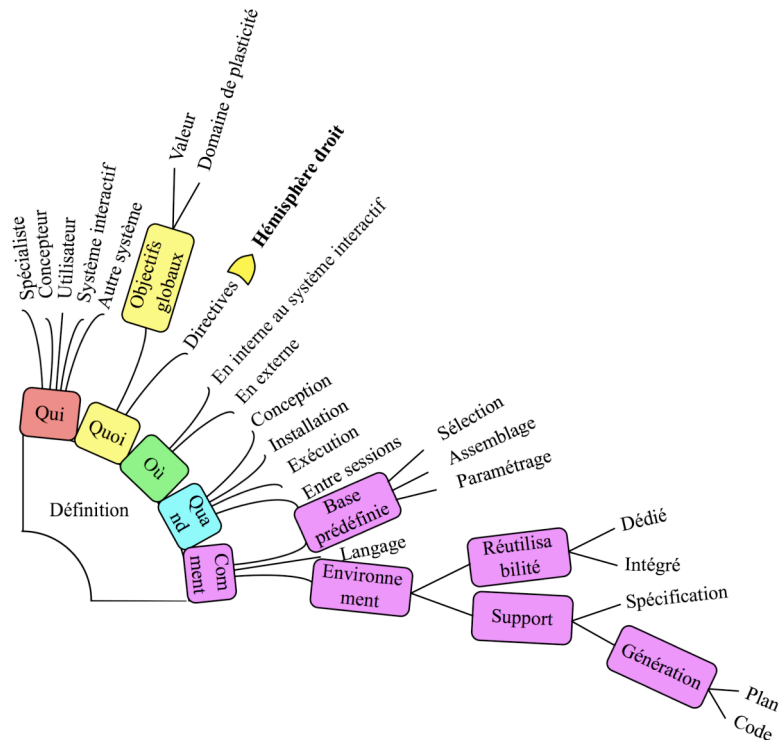


Figure 20 : Hémisphère gauche – Partie Définition.

2.2.2 Exécution

L'adaptation est un processus en cinq étapes comprenant (*Quoi*) la détection du changement de contexte d'usage puis, d'après [Dieterich 94], l'initiative de l'adaptation, la proposition de réaction, la décision et sa mise en œuvre (Figure 21). Ces fonctions requièrent des mécanismes logiciels qui peuvent être (*Où*) internes au système interactif ou externes, alors assurés par un autre système ou mutualisés dans une infrastructure logicielle. Les fonctions peuvent, de façon non exclusive, être assumées par quatre types d'acteurs (*Qui*) : l'utilisateur final, un autre utilisateur (un contrôleur par exemple), le système interactif ou un autre système (applicatif ou infrastructure). Nous identifions cinq moments clé pour leur exécution (*Quand*) : soit au fil de l'exécution, c'est-à-dire à tout instant, soit à des pas précis, le grain du pas pouvant aller de l'action physique à la session en passant par la tâche élémentaire et la tâche composée. Selon l'outil support, les actions passées seront perdues ou conservées : c'est la notion de *grain de reprise*.

Le grain de reprise indique la finesse avec laquelle l'utilisateur conserve le bénéfice de ses actions passées. Nous identifions quatre grains. Ils s'échelonnent de l'action physique (l'utilisateur conserve toute action passée) à la session (l'utilisateur perd le bénéfice de toutes ses actions passées). En grains intermédiaires, se trouvent les niveaux tâches élémentaires (les feuilles d'un arbre des tâches) et tâches composées (tâches abstraites, intermédiaires entre la racine et les feuilles dans un arbre des tâches). Le grain de reprise a une incidence directe sur la charge de travail. Aussi, si une charge de travail minimale contribue à la valeur du système interactif, alors ce grain de reprise est un point sensible de l'adaptation.

Les outils support (*Comment*) peuvent être génériques, réutilisables ou au contraire, intégrés au système interactif et, en conséquence, non réutilisables. Ils peuvent être dotés d'une IHM assurant l'observabilité du processus d'adaptation, voire son contrôle. Ils peuvent requérir des ressources (par exemple, des composants IHM, des services de perception, des agents, etc.). Ces ressources peuvent être préfabriquées ou générées à l'exécution, requérant en

conséquence des services de génération. Elles peuvent être internes ou externes au système interactif ; pérennes au-delà de l'adaptation ou, au contraire, temporaires.

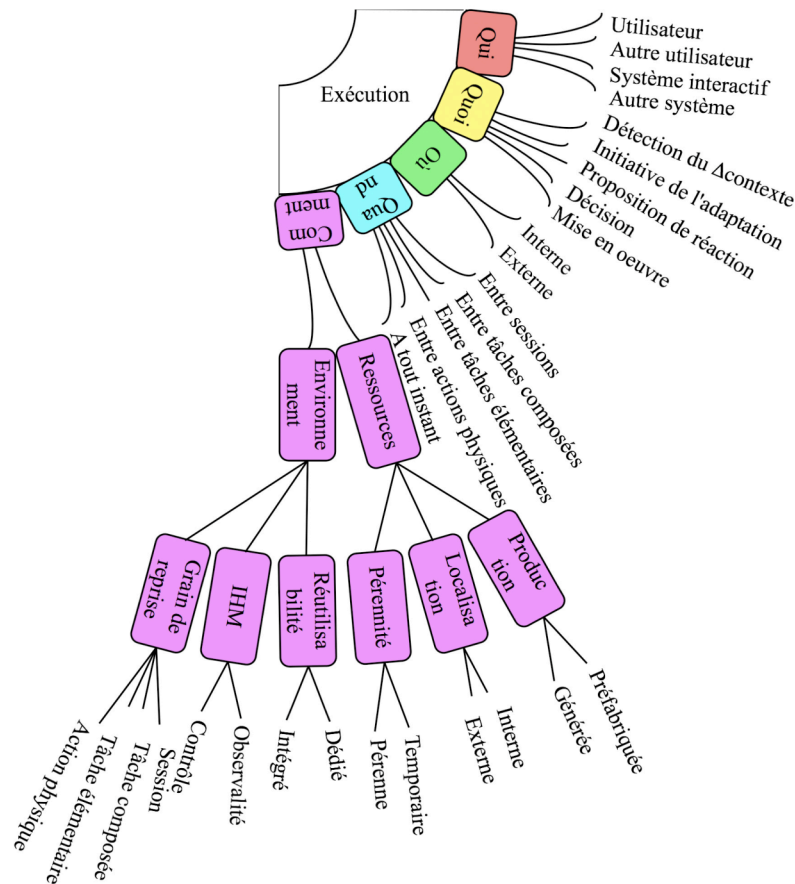


Figure 21 : Hémisphère gauche – Partie Exécution.

2.2.3 Evaluation

Autant les notions d'adaptabilité et d'adaptativité [Totterdell 90] [Browne 90] ont été largement disséminées, autant la notion d'évaluation reste discrète. Elle apparaît pourtant explicitement dans la taxonomie de [Totterdell 90]. L'hémisphère gauche lui consacre volontairement un temps particulier (Figure 22).

Nous identifions cinq acteurs (*Qui*) susceptibles d'intervenir dans l'évaluation : un spécialiste de l'évaluation, le concepteur, l'utilisateur final, le système interactif pour peu que ce dernier soit doué de capacité d'auto-évaluation (système autorégulateur selon [Totterdell 90]) ou tout autre système (applicatif ou infrastructure).

L'évaluation peut porter sur les quatre temps du cycle de vie, à savoir la définition des consignes d'adaptation, leur exécution, leur évaluation (ce qui serait alors une critique récursive) et la capitalisation d'expérience. Chaque point peut être évalué conformément aux ramifications de l'espace problème.

L'évaluation peut être réalisée (*Où*) en milieu naturel (« in the wild ») ou contrôlé « in the zoo ». Elle peut être menée à la volée ou en différé (*Quand*). Il existe différentes techniques d'évaluation. Tous les acquis sont ici applicables pour affiner la dimension du *Comment*. De façon non limitative, nous mettons en exergue deux dimensions : l'environnement support, d'une part ; la nature des résultats, d'autre part. Pour l'environnement, nous distinguons les environnements dédiés des environnements intégrés. Nous nuancions aussi selon le type de

support apporté en distinguant, conformément à [Totterdell 90], les évaluations a posteriori (système autorégulateur) des évaluations a priori (système automédiateur). En termes de résultats, nous reprenons les taxonomies classiques opposant les évaluations prédictive et expérimentale ; quantitative et qualitative ; objective et subjective. L'espace n'est pas ici exhaustif. Il mériterait d'être complété par des spécialistes en évaluation.

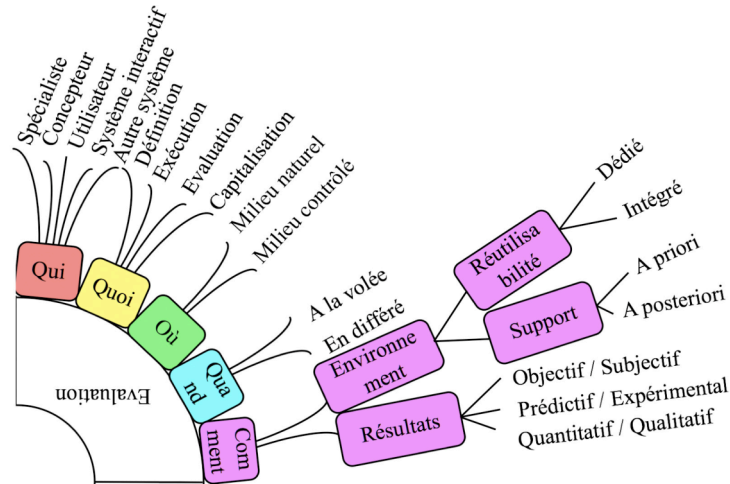


Figure 22 : Hémisphère gauche – Partie Evaluation.

2.2.4 Capitalisation

La capitalisation (Figure 23) est une dimension nouvelle qui sous-tend en particulier la prédéfinition des directives d'adaptation (cf l'axe *Base prédéfinie* du *Comment* du temps *Définition*). Cinq acteurs sont susceptibles d'intervenir (*Qui*) : un spécialiste de l'évaluation ou de la capitalisation, le concepteur, l'utilisateur final, le système interactif ou un autre système (applicatif ou infrastructure). La capitalisation peut être générale couvrant la valeur des directives quelle qu'elle soit (faible ou forte) ou se limiter aux seules directives de valeur (*Quoi*). Les expériences négatives sont alors perdues.

La capitalisation requiert des mécanismes logiciels (*Comment*). Ces mécanismes peuvent être généraux, mis à disposition dans des outils dédiés ou, au contraire, embarqués dans le système interactif lui-même. Le support offert peut aller d'une simple mémorisation des expériences à l'inférence de nouvelles directives par exemple. Les résultats peuvent être organisés ou non ; justifiés ou non. Ils peuvent être stockés de façon interne ou externe au système interactif (*Où*) ; capitalisés à la volée ou en différé (*Quand*).

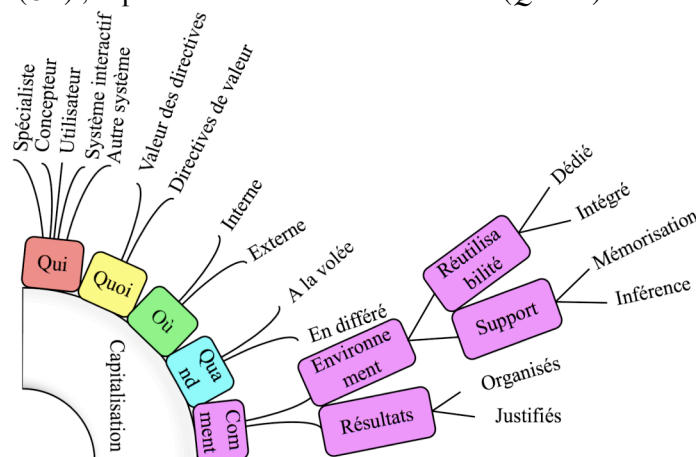


Figure 23 : Hémisphère gauche – Partie Capitalisation.

Hémisphères n'est pas le seul espace problème en matière d'adaptation. Il se veut fédérateur d'espaces existants et de compréhensions récentes. La section suivante montre son caractère fédérateur par rapport à deux espaces : [Thevenin 01] et [Vanderdonck 05].

2.3 Hémisphères, un espace problème fédérateur

[Thevenin 01] est, à notre connaissance, le premier espace problème établi pour l'ingénierie d'IHM plastiques. La figure 24 le projette sur les deux hémisphères. Toutes les dimensions y trouvent place à l'exception de deux, hors sujet pour Hémisphères : les outils support à la génération d'IHM multicibles et l'acteur à la conception. Ce n'est pas le propos ici.

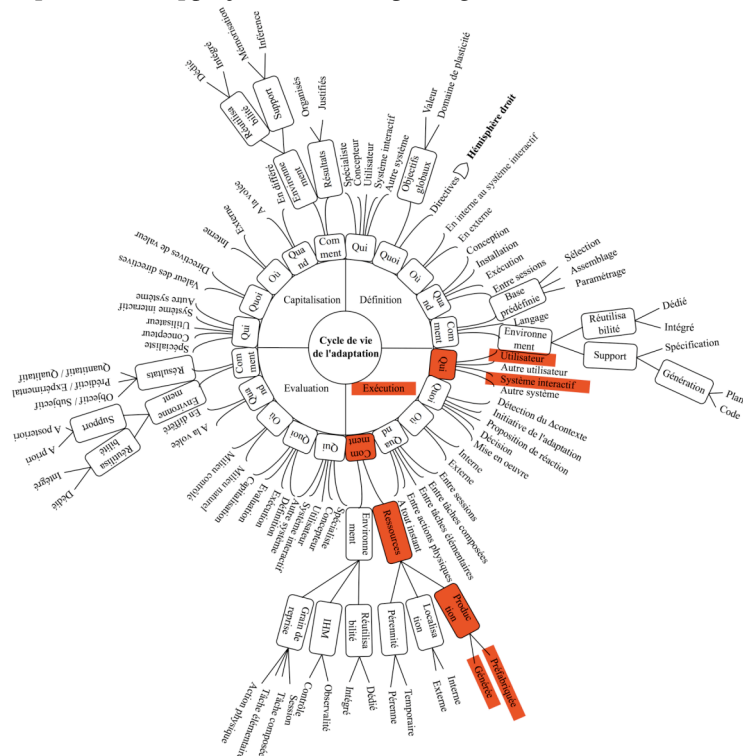
La figure 25 se prête au même exercice avec [Vanderdonck 05]. Une dimension de cet espace n'est pas couverte par Hémisphères : les modèles actifs/passifs ou mixtes. Cet aspect relève, de notre point de vue, de l'espace solution. Par ailleurs, dans la cible de l'adaptation, nous n'intégrons pas les notions de tâches, domaine et objet d'interaction qui, de notre point de vue, ne relèvent pas du contexte d'usage. Dans Hémisphères, la notion de plate-forme reste générale mais couvre en particulier les capacités de calcul et dispositifs physiques tels que détaillés dans [Vanderdonck 05]. L'« organisation » introduite dans [Vanderdonck 05] sera probablement à considérer. Mais son intégration nécessite une réflexion quant à son positionnement par rapport à la notion de contexte d'usage. Nous différons donc cet aspect. Enfin, dans la partie réaction, nous n'intégrons pas explicitement le nombre d'opérations. En revanche, le caractère atomique versus composé de la réaction est explicite.

Des figures 24 et 25, on peut tirer plusieurs enseignements :

- Une couverture aujourd'hui hybride des espaces problèmes en termes d'analyse des directives d'une part et de cycle de vie d'autre part. Ces préoccupations sont désormais séparées en deux hémisphères donnant ainsi l'opportunité d'une analyse plus fouillée de chacune d'elle ;
- un barycentre historiquement porté sur le remodelage ;
- l'existence de problèmes encore peu explorés comme la valeur, l'évaluation et la capitalisation.

Le chapitre suivant présente notre approche scientifique pour apporter une solution au problème ainsi mieux compris. Elle s'appuie sur les modèles actifs tels qu'évoqués dans [Vanderdonck 05].

a) [Thevenin 01] projeté sur l'hémisphère gauche



b) [Thevenin 01] projeté sur l'hémisphère droit

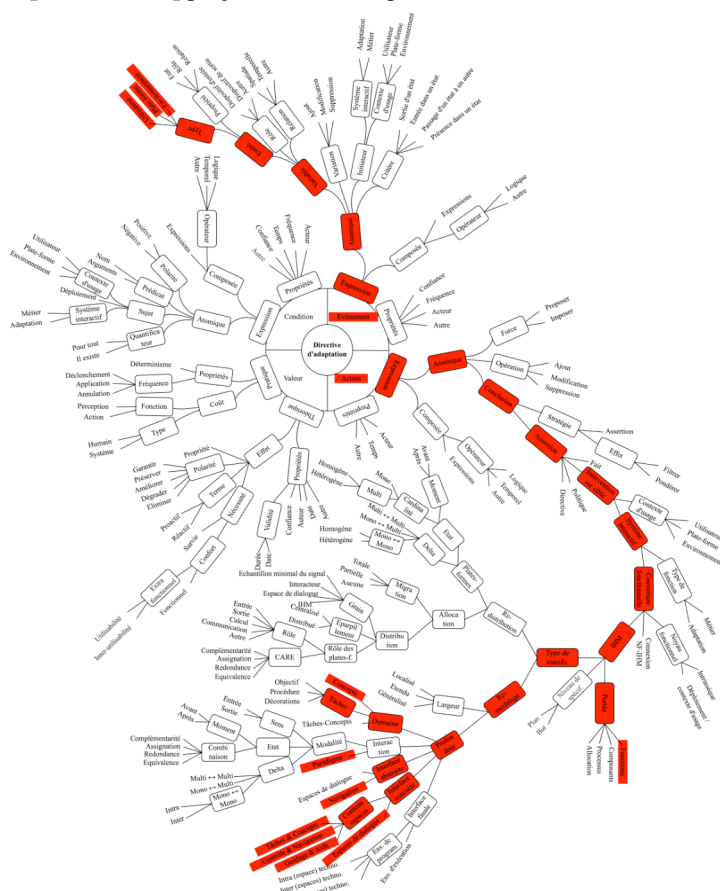


Figure 24 : Caractère fédérateur d'Hémisphères : projection de [Thevenin 01] sur Hémisphères.

Le diagramme illustre le cycle de vie de l'adaptation (CVA) sous la forme d'un cercle central entouré de trois phases principales : Définition, Exécution et Évaluation. Chaque phase est subdivisée en sous-phases, qui sont à leur tour liées à des activités ou des concepts spécifiques.

- Phase Définition :**
 - Qui** : Conception, Installation, Paramétrage, Sélection, Assemblage, Paramétrage.
 - Quoi** : Mise à jour, Prédictibilité, Langage, Environnement, Utilisateur, Autre utilisateur, Système interactif, Autre système.
 - Comment** : Détection du contexte, Proposition de l'adaptation, Décision, Mise en œuvre.
- Phase Exécution :**
 - Qui** : Externe, Interne, Entre-sessions, Entre-actes, Entre-actes physiques, Entre-actes numériques.
 - Quoi** : Ressources, Environnement, Réaction, Pervénir, Réutilisation, IHM, Granit de copier.
 - Comment** : Capitalisation, Définition, Système interactif, Conception, Conçeur, Système interactif, Autre système interactif, Autre système.
- Phase Évaluation :**
 - Qui** : Externe, Interne, Entre-actes, Entre-actes physiques, Entre-actes numériques.
 - Quoi** : Résultats, Objectif / Subjectif, Quantitatif / Qualitatif, Support, A priori, Réutilisation, Intégrité, Déclat.
 - Comment** : Spécialiste, Conçeur, Système interactif, Utilisateur, Autre système, Valeur des directives, Valeur de valeur, Mécanisme.

Le cycle de vie de l'adaptation est également lié à des concepts plus généraux tels que : Capitalisation, Définition, Exécution, Évaluation, et des termes comme "Hémisphère droit" et "Hémisphère gauche" qui pourraient indiquer une dimension cognitive ou organisationnelle.

44

Chapitre 2. Une approche scientifique pour agir



*La plasticité, un sujet en trois temps.
Deuxième mesure : définir une approche pour agir.*

L'adaptation n'est pas un problème nouveau. Il n'est toujours pas réglé. Il s'est aujourd'hui complexifié avec, d'une part, la possibilité de redistribuer les IHM sur un ensemble de plates-formes et, d'autre part, l'imprévu qui prévaut en informatique ambiante. Il est donc important de comprendre pourquoi les approches jusqu'ici mises en œuvre n'ont pas permis de le résoudre.

De notre compréhension, le problème est complexe (cf Hémisphères) et les travaux restent cloisonnés par discipline et au sein même de chaque discipline. Entre disciplines, on distingue deux angles d'attaque :

- d'une part, l'adaptation avec un flambeau porté en Génie Logiciel par IBM avec l'Autonomic Computing [Kephart 03]. D'autres communautés y travaillent, en particulier, l'IHM [Thevenin 01] [Paganelli 02] [Calvary 03] [Limbourg 04a] [Berti 05] [Florins 06] [Daassi 07] ;
- d'autre part, l'information située [Suchman 87] et la technologie support dite « context aware computing » [Dey 00]. Dans cette technologie support⁹, [Tigli 06] classe les travaux selon la propension donnée aux différents aspects du contexte. Il distingue les intergiciels « sensibles au contexte » des intergiciels « sensibles aux ressources ». Les intergiciels sensibles au contexte sont centrés sur l'utilisateur et son environnement physique [Capra 03]. Ils ne sont généralement pas spécifiques à un domaine d'application. Ils considèrent souvent la localisation des utilisateurs [Baldauf 04], leurs préférences et certaines conditions [Bellavista 06] pour déterminer les ressources accessibles en fonction des services souscrits. Les intergiciels sensibles aux ressources [Mahéo 04] exploitent essentiellement les propriétés dynamiques de l'infrastructure système. Ils considèrent aussi bien des caractéristiques physiques (dispositifs de stockage, communication, interaction, etc.) que logiciels (bibliothèques et services divers).

Si des tentatives de rapprochement sont amorcées (pour exemple, en local, le projet IMAG MAPPING ; en national, l'Action Spécifique Plasticité du CNRS ou encore l'axe de travail CESAME du thème 5 du GDR I3), les rapprochements restent essentiellement conceptuels sans que, d'un point de vue logiciel, les travaux ne soient intégrés.

⁹ Informations de Jean-Yves Tigli, Stéphane Lavirotte et Anne-Marie Déry. Extrait d'une proposition commune. Merci à eux.

Au sein des disciplines, les travaux sont aussi cloisonnés. La partition se fait selon l'objet étudié ou le paradigme suivi. En IHM, [Demeure 07] distingue trois classes d'approches pour l'adaptation : l'Ingénierie Dirigée par les Modèles (IDM) ; les boîtes à outils (BàO) d'interacteurs plastiques et enfin les approches de type Operating System (OS) touchant au système d'exploitation des plates-formes cibles. La Figure 26 situe ces approches selon le niveau d'abstraction du système interactif qu'elles manipulent. Les approches IDM ont produit toute une gamme de modèles de niveaux d'abstraction divers allant du domaine à l'interface finale [Calvary 03]. Dans une approche descendante, le code est généré par transformation de ces modèles : la transformation est alors dite de type « Concrétisation » ou « Réification ». Inversement, les modèles peuvent être découverts à partir du code : la transformation est alors dite de type « Abstraction ». En intermédiaire, dans les approches de type BàO, l'IHM est construite par assemblage d'interacteurs. Là encore, on distingue les BàO selon leur dominance *polymorphisme* (ACE [Johnson 93], WAHID [Jabarin 03], XFORMS du W3C), *multimodalité* (FRUIT [Kawai 96], Multimodal Widgets [Crease 00]) ou *post-WIMP* (UBIT [Lecolinet 03], Attach Me Detach Me [Grolaux 05]). Enfin, les approches de type OS agissent au niveau du système d'exploitation (par exemple, le système de fenêtrage), produisant en conséquence une plasticité non limitée au seul système interactif étudié. C'est le cas de Facade [Stuerzlinger 06].

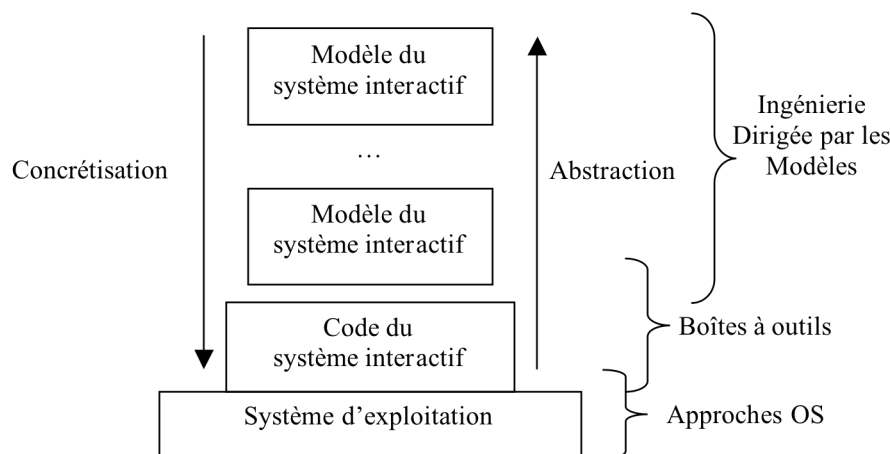


Figure 26 : Positionnement des trois classes d'approches (IDM, BàO et OS) identifiées par [Demeure 07] en adaptation des IHM.

Historiquement, les premiers travaux en plasticité se situent dans le champ de l'IDM. Il est intéressant de noter que les approches étaient alors dites « basées modèles ». C'est l'influence du Génie Logiciel qui leur a valu d'être rebaptisées. Initialement portée par l'OMG sous le drapeau du MDA (Model Driven Architecture), cette initiative avait comme point de départ la séparation des modèles selon qu'ils étaient indépendants ou dépendants des plates-formes (Platform Independent Model ou PIM versus Platform Specific Model ou PSM). La notion de plate-forme faisait principalement référence à des intergiciels comme EJB ou DotNet. Depuis, l'OMG a abandonné l'idée de la plate-forme unique ou unifiable. Au contraire, il s'agit désormais d'être capable de capitaliser le savoir-faire indépendamment des plates-formes, de le projeter dans différents espaces technologiques et de naviguer de l'un à l'autre selon la capacité de ces espaces à régler le problème traité [Sottet 07c].

De même, en système¹⁰, plusieurs angles d'attaque s'affirment. Différents paradigmes logiciels se dégagent pour l'adaptation : la réflexivité logicielle [Maes 87], la conception

¹⁰ Informations de Jean-Yves Tigli, Stéphane Lavirotte et Anne-Marie Déry. Extrait d'une proposition commune. Merci à eux.

orientée composant [Szyperski 98] et les travaux sur les ADL (Architecture Description Language), schémas de conception [Schmidt 00]. Les approches orientées Composants comme les approches orientées Services (SOA) [Schulte 96] sont dirigées par le même principe de « séparation des préoccupations » [Lopes 95]. Ce principe est à la base même de la Programmation par Aspects [Kiczales 97]. Des outils appelés Tisseurs (Weavers) produisent, à partir d'une spécification de points d'entrelacement, le code « mêlant » les parties tissées. La programmation par aspects est utilisée pour l'adaptation (par exemple, [Rashid 04]). Une version adaptable des composants Fractal est même proposée [David 06]. Les composants peuvent être reconfigurés dynamiquement par des aspects en utilisant des stratégies d'adaptation *Event-Condition-Action*. En France, l'équipe RAINBOW, en particulier, travaille ardemment sur le sujet.

Au-delà du spectre fin des travaux, nous repérons comme faille une dichotomie inopportune entre les phases de conception et d'exécution : Hémisphères montre clairement une frontière de plus en plus ténue entre les rôles de concepteurs et d'utilisateurs. L'approche scientifique que nous préconisons (MUST pour Mixing & Unifying Systems & Transformations) brise cette frontière : elle *unifie* conception et exécution autour de deux concepts fondamentaux, les systèmes et transformations. Ces concepts sont définis en IDM. Les transformations sont opérées par des services. MUST *mixte* ainsi un ensemble d'acquis pour nous donner toutes les chances de couvrir Hémisphères.

La première section présente les deux principes directeurs de MUST : l'unification et la mixité. Ils sont étayés, dans la section suivante, de deux concepts fondateurs : les Systèmes et Transformations. La troisième section développe MUST sous la forme de dix principes.

1. Principes directeurs de MUST : Unification et Mixité

En point de départ, nous posons comme acquis et applicable notre savoir-faire en ingénierie d'IHM traditionnelles. Nous ne révolutionnons pas les méthodes, modèles et outils pour traiter la plasticité. Nous les augmentons pour couvrir les dimensions vierges de l'espace problème. La force et l'originalité de l'approche viennent :

- d'une unification des travaux entre la conception et l'exécution d'IHM. L'hémisphère gauche est éloquent : il montre la banalisation des acteurs humain et système dans les phases du cycle de vie. Avec l'informatique ambiante, les contextes d'usage ne sont plus tous prévisibles à la conception. Il devient donc, en particulier, nécessaire, à l'exécution, de concevoir des IHM aptes à couvrir le contexte d'usage courant dans le respect de la valeur attendue. C'est l'unification entre la conception et l'exécution ;
- d'une mixité des approches scientifiques et technologiques pour une souplesse méthodologique et un renforcement des approches. En particulier, en IHM, il s'agit de réunir autour d'une même infrastructure les concepteurs qu'ils soient enclins modèles ou « code ». L'écosystème doit être hybride, tissant les modèles dans le code exécutable/interprétable. Il doit s'ouvrir sur des compétences hors IHM. Nous renforçons ici le Génie Logiciel de l'Interaction Homme-Machine par des apports de l'Intelligence Artificielle pour permettre au système de raisonner sur sa propre conception à l'exécution et apprendre au fil de l'interaction.

Nous développons ces deux principes directeurs dans les sections suivantes.

1.1 Unification

Il existe traditionnellement une dichotomie forte entre les phases de conception et d'exécution. Sur la base d'un cahier des charges, le concepteur sait, par une série de

compromis, concevoir une IHM de qualité. Il le saura d'autant plus qu'il a de l'expérience. En cas de doute, il mettra l'utilisateur dans la boucle lui soumettant, par exemple, différentes versions possibles d'IHM : les conceptions participatives ont fait leur preuve en IHM [Schuler 93]. En revanche, ce savoir-faire mobilisé dans le processus de conception est volatile : il n'est pas consigné dans le produit résultant. Ainsi, à l'exécution, une IHM ignore la raison d'être de chacun de ses éléments et est donc incapable de soutenir tout raisonnement quant à son adaptation. Prenons l'IHM de la figure 27a inspirée de [Nogier 05]. Les trois libellés *Nom*, *Prénom* et *Date* ont une valeur de « Guidage-Incitation » [Bastien 93] par rapport aux champs texte qu'ils annoncent. Donc si la fenêtre était rétrécie, d'après le critère de « Guidage-Groupement/distinction entre items par la position » [Bastien 93], ces libellés devraient rester groupés avec les champs texte correspondants. Les figures 27b et c proposent deux adaptations possibles au regard de ce critère. En revanche, l'alternative de la figure 27d transgresserait ce critère. Il serait surprenant qu'un concepteur imagine une telle conception. Pourtant, aujourd'hui rien n'empêcherait le système d'imaginer cette transformation, surtout si, pour des raisons techniques d'alignement entre libellés et champs texte par exemple, le programmeur a créé deux canevas groupant à gauche les libellés et à droite les champs texte. On sent ainsi la pertinence d'embarquer dans les IHM la raison d'être de chacun de leurs éléments : c'est le concept d'*IHM autoconnaissante*.

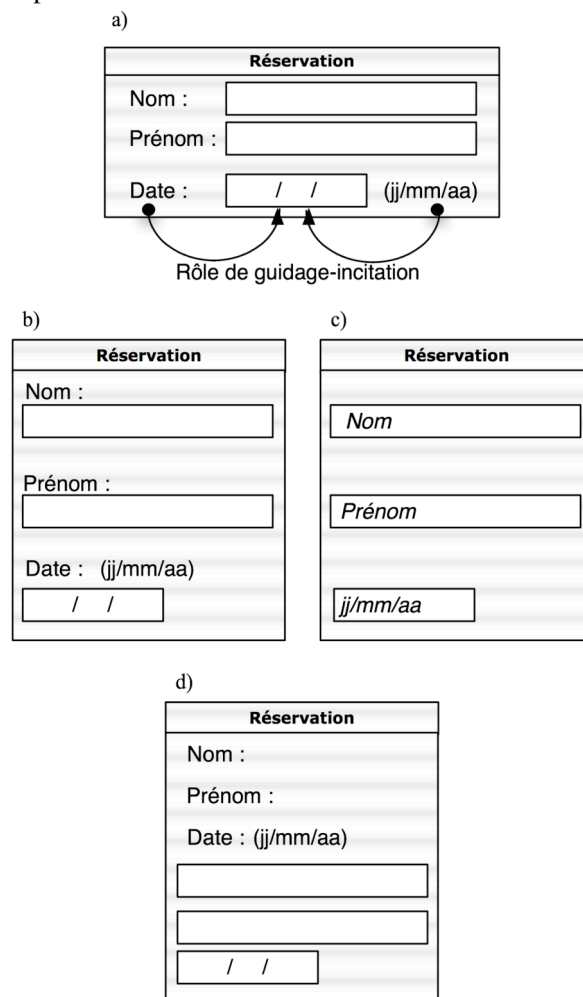


Figure 27 : En a), une IHM « grande » largeur. En b) et c), deux transformations possibles. En d), une transformation à proscrire parce qu'ergonomiquement insatisfaisante.

Une IHM autoconnaissante est une IHM qui connaît sa raison d'être. Chaque interacteur sait dans quel espace de dialogue il se trouve et à quelle tâche il correspond. Cette autoconnaissance règle une difficulté majeure en évaluation expérimentale : remonter des actions physiques aux tâches utilisateur [Calvary 98]. Si l'IHM est capable de communiquer

sur cette connaissance, alors elle est dite *auto-explicative*. Le concept du « Why and Why not » en relève [Myers 06]. L'auto-explication peut servir différents acteurs :

- un concepteur qui arriverait sur le projet et serait en charge du maintien en conditions opérationnelles de l'IHM ;
- l'utilisateur final qui, grâce à une lentille magique (Figure 28) ou d'autres Extra-IHM, accéderait ainsi à une aide contextualisée ;
- l'IHM qui voudrait se transformer (ou le système la transformer) pour couvrir un contexte d'usage non prévu à la conception.

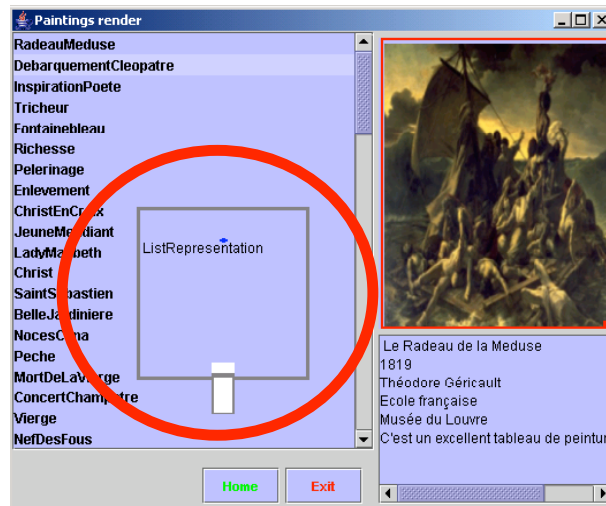


Figure 28 : Une lentille magique permet d'explorer les dimensions cachées de l'interacteur. Elle en affiche ici la fonction de service, à savoir la représentation d'une liste. Extrait de [Dâassi 07].

Concepteurs, utilisateurs et systèmes manipuleraient alors un même objet, une IHM autoconnaissante, qui selon le rôle de son « manipulateur » (concepteur, utilisateur, système) saurait se présenter et se laisser manipuler (Figure 29).



Figure 29 : L'IHM, un objet polymorphe. Tâches, concepts, espaces de dialogue, etc. sont autant de perspectives différentes sur une même IHM. Dessin de Serge Cecconi.

Nous appelons *polymorphisme* l'existence de plusieurs « formes » (c'est-à-dire ici présentations). Nous appelons *aristomorphisme*¹¹ la capacité d'une IHM à choisir la forme la plus appropriée selon le contexte d'usage. En particulier, pour le concepteur, l'IHM se présenterait selon les différents points de vue que nous savons aujourd'hui pertinents :

- Domaine : il verrait le modèle des tâches, le modèle des concepts, le lien entre les deux modèles (cf chapitre I) ;
- Les espaces de dialogue et la navigation ;
- Les interacteurs tels que perçus par l'utilisateur final.

Bien entendu, pour les niveaux amont (domaine et interface abstraite), il s'agira d'inventer les "bons" interacteurs. Prenons, par exemple, le cas des tâches. L'IHM de manipulation d'un modèle des tâches est l'IHM qui correspond à la tâche *Editer modèle de tâches*. Elle manipule un concept qui est un modèle des tâches. Dans le respect du critère de « Compatibilité par rapport aux autres environnements » [Bastien 93], on pourrait s'inspirer d'éditeurs tels que CTTE [Mori 02] pour l'IHM. C'est ce qui a été fait dans de nombreux environnements tels qu'ArtStudio [Thevenin 01]. Mais peut-être d'autres présentations sont-elles pertinentes. A noter qu'offrir ces perspectives en modification à l'utilisateur final, c'est couvrir le « end-user programming » (Figure 30).

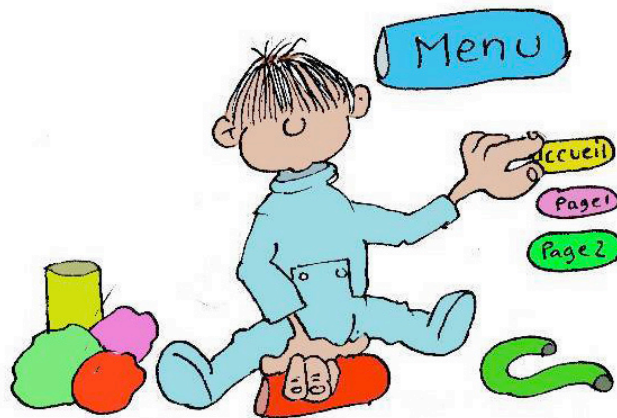


Figure 30 : End-user programming : l'utilisateur modèle son IHM à la pâte à modeler. Dessin de Serge Cecconi.

En conclusion, unifier la conception et l'exécution, c'est concentrer les recherches sur un même objet d'étude : l'IHM autoconnaissante. Pour cette étude, nous préconisons une approche ouverte, mixant les avancées de différentes communautés.

1.2 Mixité

Mixer les approches aussi bien scientifiques que technologiques, c'est nous donner toutes les chances de :

- Gagner en recul, souder les communautés ;
- Disposer de solutions affûtées pour des problèmes précis, pallier les faiblesses d'une approche par une autre. On se souvient de la corrélation de techniques en vision par apparence [Martin 95] ;
- Respecter le savoir-faire humain. Lorsque nous sommes performants dans un langage donné, avec un outil donné, il est toujours désagréable, déstabilisant et, en conséquence, source de non performance, de changer d'outil.

¹¹ Merci à Christian Boitet.

[Demeure 07] propose, par exemple, un style d'architecture logicielle (les COMET pour CONtext Mouldable WidgETs) qui réconcilie les approches IDM et BâO : les facettes des interacteurs se calquent sur les niveaux d'abstraction mûris dans les approches orientées Modèles. Les COMET sont, par ailleurs, multi-technologies : leur rendu est assuré aussi bien dans le monde Web (AJAX) que non Web (des BâO d'utilité publique telles que TK ou spécifiques telles que B207 pour le post-WIMP ou S207 pour le vocal), alors qu'une dichotomie est traditionnellement faite entre ces deux mondes. La figure 32 montre des exemples de rendu pour la télécommande de CamNote. Ces rendus sont activables simultanément.

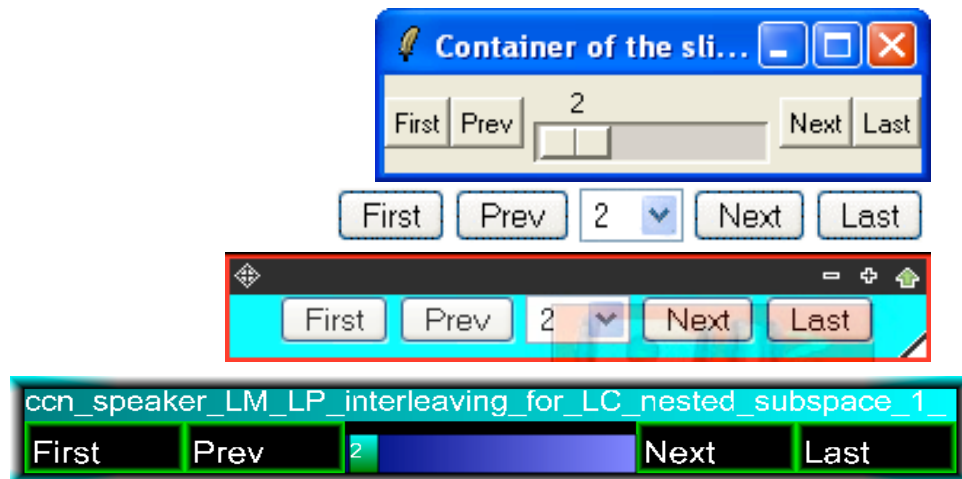


Figure 31 : Exemples de rendus TK, HTML et B207 pour la télécommande CamNote en COMET.
Extrait de [Demeure 07].

[Demeure 07] propose, par ailleurs, un point d'ancrage de la multimodalité dans les COMET ainsi qu'un langage pour le traitement de la redondance et de l'équivalence (COMET/RE). Une ouverture est également faite sur les annuaires de services (GDD) et les langages de style (CSS++).

Une autre initiative est l'infrastructure logicielle MARA (Models At Runtime for Adaptation) qui, par une approche à Services, voit les outils de perception du contexte d'usage et de transformation d'IHM comme des services.

Ces deux initiatives sont décrites dans le chapitre 3. Elles nourrissent notre approche MUST. Elles ne sont pas encore intégrées mais la clé de voûte réside dans les notions de systèmes et de transformations.

2. Concepts fondateurs de MUST : Systèmes et Transformations

Après le « Tout est objet » des années 80, tout est devenu Modèle avec l'IDM. Aujourd'hui, tout devient Système, la notion de modèle n'est plus que relative : un modèle est un système qui joue un rôle particulier (celui de modèle) vis-à-vis d'un autre système. Par exemple, une mappemonde est un *système* qui peut jouer le rôle de *modèle* pour un autre *système* : la planète Terre en l'occurrence. Mais elle ne jouera ce rôle de modèle que pour une personne établissant ce lien de *représentation* entre les deux instances [Sottet 05a]. De la même façon, le dessin de la figure 32a sera interprété de différentes manières selon le lecteur :

- Comme deux cercles tangents à un troisième pour un mathématicien ;
- Comme un agent PAC pour un spécialiste de l'architecture logicielle des systèmes interactifs ;
- Ou comme un *Mickey* pour une personne un peu plus fantaisiste.

Clairement les représentations des figures 32b et 32c limiteraient les ambiguïtés d'interprétation.

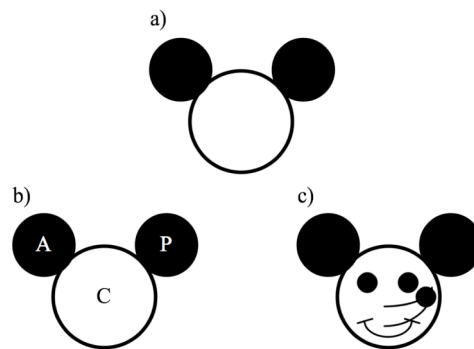


Figure 32 : Trois modèles... mais de quels systèmes ?

En IDM, il existe fondamentalement quatre concepts et quatre relations clé [Sottet 05b] :

- Les Systèmes et la relation *SeDécomposeEn* (δ).
En IDM, la notion de système est le premier élément du discours : tout est système. Un système peut être simple ou complexe. Un système complexe est composé d'autres systèmes : c'est la relation *Se décompose en*. Elle est notée δ . Par exemple, le système solaire se décompose en planètes ;
- Les modèles et la relation *Représente* (μ).
Il n'existe pas de définition universelle pour le concept de modèle. Cependant, on peut trouver un consensus relatif autour du fait que *modèle* et *système étudié* sont deux rôles complémentaires. La relation qui lie un modèle à un système étudié est notée μ . De manière simplifiée, on peut dire qu'un modèle est une représentation simplifiée d'un système, représentation utilisée pour répondre à des questions à la place de ce système [Favre 04a] ;
- Les métamodèles et la relation *EstConformeA* (χ).
Un métamodèle est un modèle d'un langage de modélisation. Le concept de métamodèle mène à la relation notée χ souvent appelée *EstConformeA*. Cette relation permet de décrire le lien entre un modèle et le métamodèle modélisant le langage dans lequel il est décrit. Dans le cas particulier de la technologie orientée objet, la relation *InstanceDe* correspond à cette notion de conformité, mais ce n'est là qu'un cas particulier. Par exemple une phrase peut être conforme (ou non) à une grammaire. L'article [Favre 04b] est dédié à l'étude de cette relation χ ;
- Les transformations et la relation *EstTransforméEn* (τ).
Mettre l'accent sur la relation χ permet d'envisager des transformations automatisées d'un modèle vers un autre. La relation τ introduite dans [Favre 04c] et [Favre 04d] relie deux modèles. Le coeur de l'IDM consiste à représenter de manière explicite les transformations donnant lieu à la notion de modèle de transformation. Ces modèles s'appuient sur les métamodèles des modèles que l'on désire transformer [Favre 04d]. La relation *EstTransforméEn* consiste donc à prendre des éléments d'un ou plusieurs modèles et de les mettre en correspondance avec d'autres éléments de modèle(s). Cette mise en correspondance provient d'un savoir-faire métier. Les transformations de modèles sont une manière de capitaliser le savoir-faire.

La notion de *mégamodèle* [Favre 04c] a été introduite pour rendre explicites les entités et relations manipulées dans l'IDM (Figure 33). Cette notion, même si elle n'apporte rien de nouveau, est précieuse pour une compréhension rapide et globale de l'IDM.

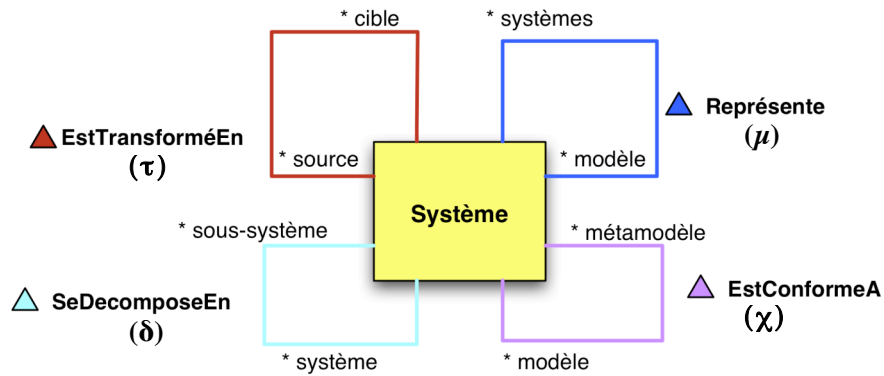


Figure 33 : Méga-modèle. Extrait de [Sottet 05a].

La communauté IHM se reconnaît dans le vocabulaire IDM. Les notions de modèles et de transformations ne sont pas nouvelles en IHM. Il existe même un long savoir-faire. Les modèles sont reconnus pour leur pouvoir méthodologique, mais le carton rouge vint des générateurs d'IHM : la qualité des IHM produites était décevante.

Aujourd'hui, nous réhabilitons les modèles dans un esprit d'unification et de mixité :

- Unification : les modèles sont vivants à l'exécution. Les transformations classiquement opérées à la conception peuvent l'être à l'exécution ;
- Mixité : les approches cohabitent. Les systèmes peuvent être des composants logiciels développés à façon (par exemple, une technique d'interaction post-WIMP). Ils peuvent être du code généré à partir de modèles (par exemple, d'un modèle de tâches). Mais ils peuvent être aussi des modèles. Les systèmes sont transformables par des moteurs de transformation. Ces moteurs peuvent être des services qui se découvrent dynamiquement.

Nous énonçons, dans la section suivante, notre approche en dix principes.

3. MUST en dix principes

Dans la première section, nous présentons un cas d'étude simple en support à l'illustration. Il s'agit de HHCS (Home Heating Control System), une application de domotique.

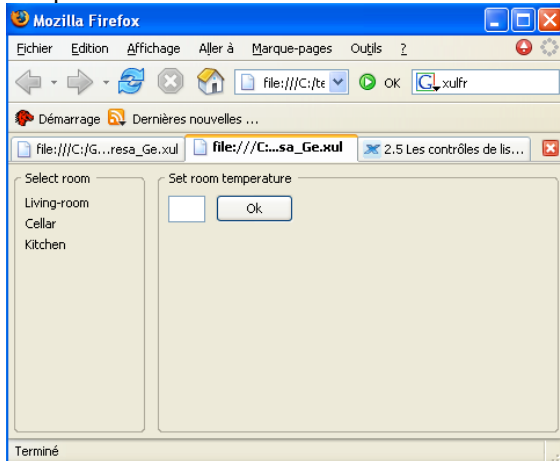
3.1 Cas d'étude

HHCS permet à un usager de contrôler la température de sa maison via différentes IHM. Les IHM diffèrent par leur état de distribution (centralisées en Figures 34a-d versus distribuée en Figure 34e) ainsi que par leurs propriétés ergonomiques :

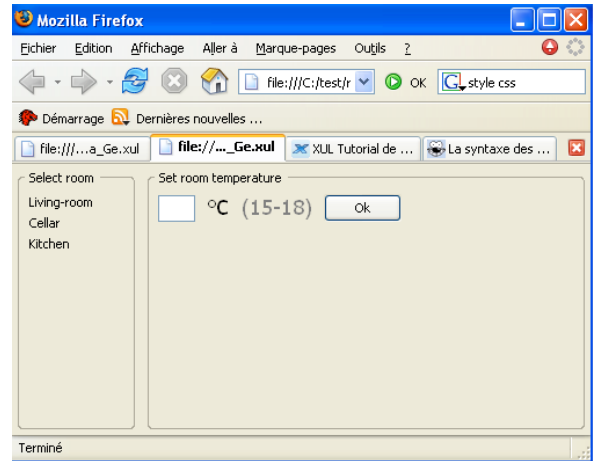
- Dans la figure 34a, ni l'unité de mesure (°C), ni les valeurs acceptables (de 15 à 18 °C) ne sont affichées. Ceci transgresse le critère de « Guidage », sous-critère « Incitation » [Bastien 93] ;
- En b, ces défauts sont réparés mais l'utilisateur n'est pas pour autant protégé contre les erreurs de saisie. Ceci transgresse le critère de « Gestion des erreurs », sous-critère « Protection contre les erreurs » [Bastien 93] ;
- En c, l'imperfection de la Figure 34b est contrée par un menu déroulant. En revanche, l'utilisateur perd toujours un clic pour sélectionner la pièce d'intérêt dans le bandeau gauche. Ceci transgresse le critère de « Charge de travail », sous-critère « Brièveté dans les actions » [Bastien 93] ;

- En d, l'anomalie est réparée mais la solution n'est pas viable pour un château de cinquante pièces au risque sinon de dégrader cette fois la charge de travail dans son sous-critère « Densité informationnelle » [Bastien 93] ;
- En e, l'IHM est distribuée : la sélection de la pièce se fait sur PDA ; le contrôle de la température sur PC. Cette répartition au grain des espaces de dialogue ne se justifie pas ergonomiquement. HHCS est un cas d'étude simple, utilisé pour raisonner.

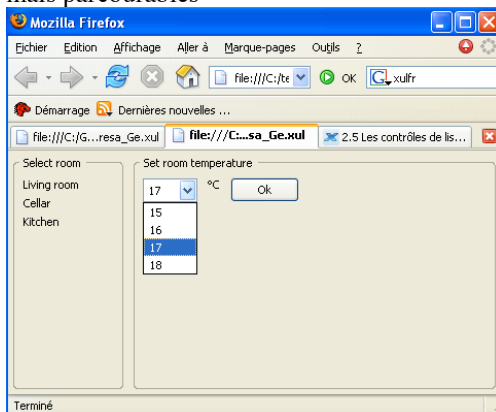
(a) Ni l'unité de température, ni les valeurs acceptables ne sont observables



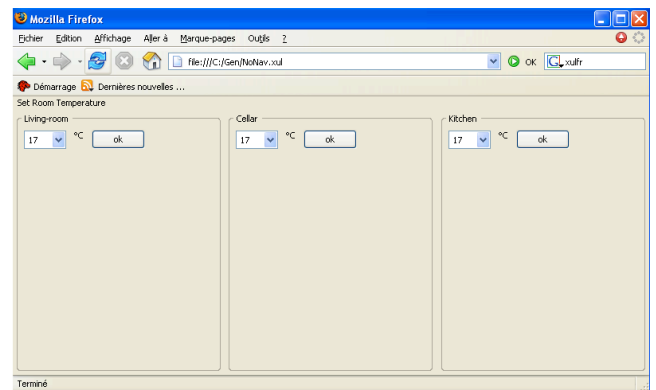
(b) L'utilisateur n'est pas protégé contre les erreurs



(c) Les pièces ne sont pas directement observables mais parcourables



(d) Les pièces sont directement observables



(e) L'IHM est distribuée entre un PC et un PDA : la sélection de la pièce se fait sur PDA, la spécification de la température sur PC

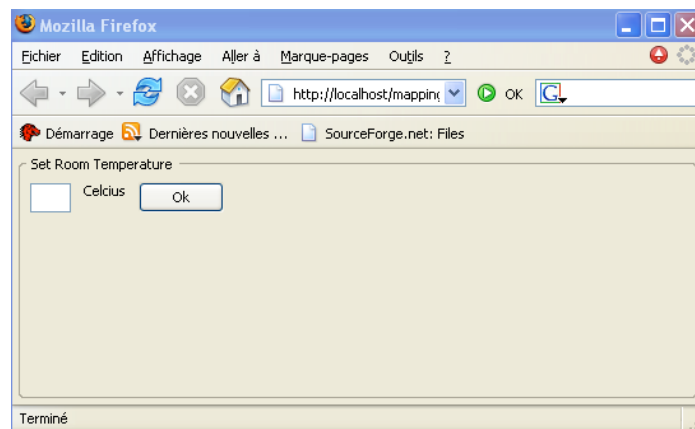
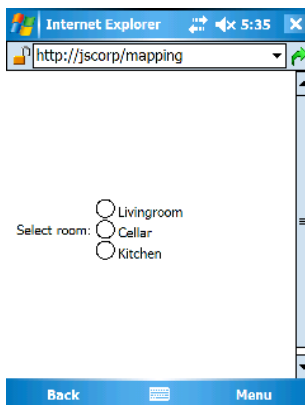


Figure 34 : Cinq versions possibles d'une même IHM selon l'état de distribution souhaité (centralisé versus distribué) et les propriétés d'ergonomie privilégiées. (X. Alvaro et JS Sottet).

Le cas d'étude étant décrit, nous développons l'approche en dix principes dans la section suivante.

3.2 Les dix principes de MUST

Principe #1 : Un système interactif en contexte est décrit par un graphe de modèles couvrant potentiellement :

- 1.a) les niveaux d'abstraction de l'IHM
- 1.b) les correspondances entre ces niveaux d'abstraction
- 1.c) le noyau fonctionnel
- 1.d) la connexion de l'IHM au noyau fonctionnel
- 1.e) le contexte d'usage
- 1.f) les relations entre constituants du contexte d'usage
- 1.g) le déploiement de l'IHM et du noyau fonctionnel sur le contexte d'usage
- 1.h) les relations avec les autres systèmes interactifs.

Alors que le cadre de référence CAMELEON était jusqu'ici utilisé comme support méthodologique pour raisonner sur les étapes de conception et chemins de conception, ces étapes définissent désormais différentes perspectives sur un même objet d'étude qu'est l'IHM. Nous reprenons les niveaux classiques, à savoir (Figure 35) : tâches, concepts, interfaces abstraite (AUI) et concrète (CUI). Nous ne considérons pas l'interface finale dont le modèle serait le code exécutable/interprétable de l'IHM. Un modèle pertinent serait celui de son architecture logicielle (par exemple, le style d'architecture qu'il respecte) mais ce point n'est pas encore traité. Pour le contexte d'usage, nous considérons les trois facettes, à savoir : l'utilisateur, la plate-forme (en particulier, ses capacités d'entrée et de sortie) et l'environnement. Nous n'affinons pas le noyau fonctionnel : ceci relèverait du Génie Logiciel.

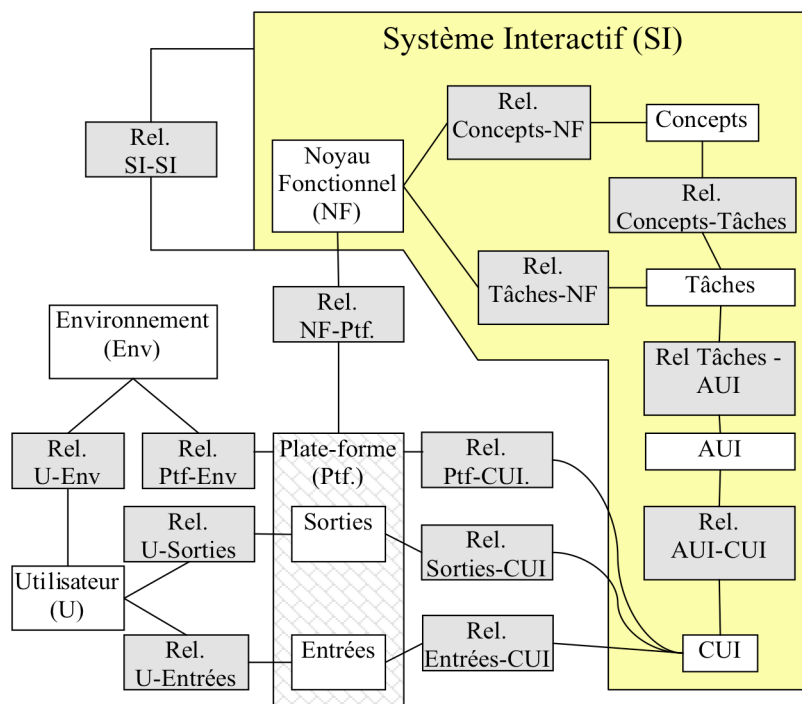


Figure 35 : Modèle de système interactif en contexte. Les boîtes (blanches et grises) représentent des modèles. Les boîtes grises sont les relations (Rel) entre modèles. Extrait de [Demeure 07].

Les modèles sont reliés entre eux (Figure 35). En particulier, les transformations entre modèles d'IHM sont conservées : cette perte d'informations était connue sous le nom du « mapping problem » [Clerckx 04] [Griffiths 01] [Limbourg 04b]. La distribution de l'IHM sur la plate-forme est également un ensemble de liens entre modèles. Redistribuer l'IHM, c'est modifier ces liens, c'est-à-dire changer les mises en correspondance entre interacteurs et plates-formes. La métaphore de la pieuvre est utilisée en Figure 36 : les interacteurs sont ventousés sur telle ou telle plate-forme pour en spécifier la distribution. L'approche unifie le remodelage et la redistribution. La position de l'utilisateur et des plates-formes dans l'environnement sont d'autres liens entre modèles. Il en est de même pour le déploiement du système intractif sur le contexte d'usage et ses éventuels liens avec d'autres systèmes interactifs.

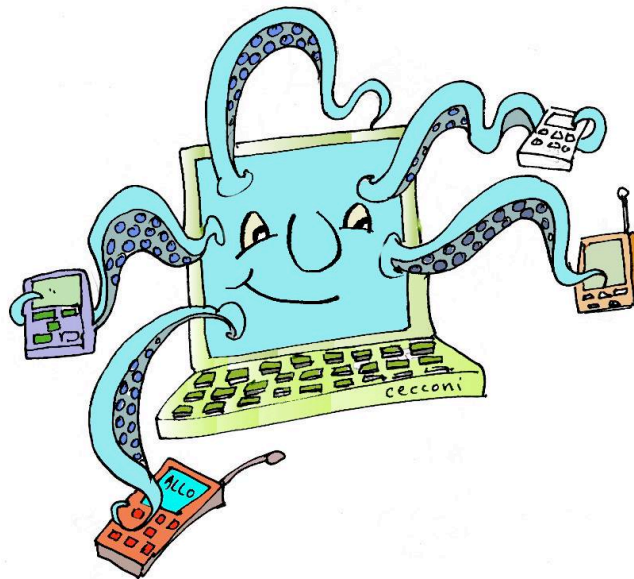


Figure 36 : Métaphore de la pieuvre pour la spécification de la redistribution. Dessin de Serge Cecconi.

Principe #2 : Les transformations sont décrites par des modèles

[Sottet 07b] clarifie les notions de lien (link), mise en correspondance (mapping), transformation et patron (pattern). Nous relatons ici ce travail sur l'exemple du cas d'étude. La Figure 34a regroupe, dans le bandeau gauche, un ensemble de liens hypertexte permettant de sélectionner la pièce d'intérêt : salon (living-room), cellier (cellar) ou cuisine (kitchen). Chaque lien hypertexte est relié à la tâche *Select room* : c'est un lien (link). L'ensemble des liens projetant la tâche *Select room* sur des interacteurs est une mise en correspondance (mapping) entre les modèles de tâches et d'interacteurs. Un mapping devrait rendre explicite sa fonction de transformation, c'est-à-dire l'ensemble des liens (link) auquel elle donne lieu. Cette description peut se faire par un modèle de transformation qui décrit alors de façon prédictive et non effective l'ensemble de ces liens. Un modèle de transformation peut être spécifié par des patrons : par exemple, appliquer au patron de tâche *Select room* le patron *Lien hypertexte dont le nom est celui de la pièce*. Recourir aux patrons est intéressant pour le critère d'« Homogénéité-Cohérence » [Bastien 93].

Les liens et les mises en correspondance devraient rendre explicites les propriétés qu'ils satisfont. Sur la figure 37, les modèles de tâches et de concepts (partie gauche) sont mis en correspondance avec le modèle d'interacteurs (ici représenté en partie droite sous sa forme

« finale » pour des raisons de lisibilité). Les transformations rendent explicites les correspondances entre :

- tâches et espaces de dialogue : la fenêtre (Mozilla Firefox) correspond à la tâche racine *Manage home temperature* ; le bandeau gauche de cette fenêtre permet la sélection d'une pièce (*Select room*) ; la partie droite permet la spécification de la température (*Set room temperature*). Ces mises en correspondance sont motivées par deux propriétés : d'une part, la « Compatibilité par rapport à la tâche » ; d'autre part, le critère de « Guidage » dans son sous-critère « Groupement/distinction entre items par la position » ;
- concepts et interacteurs : les valeurs extrêmes du menu déroulant (15 et 18) correspondent respectivement aux valeurs minimale (*minValue*) et maximale (*maxValue*) du concept de température. Le menu déroulant satisfait le critère de « Gestion des erreurs » dans son sous-critère « Protection contre les erreurs » [Bastien 93]. Le libellé « °C » est la représentation de l'unité (*unit*). C'est du « Guidage », sous critère « Incitation ».

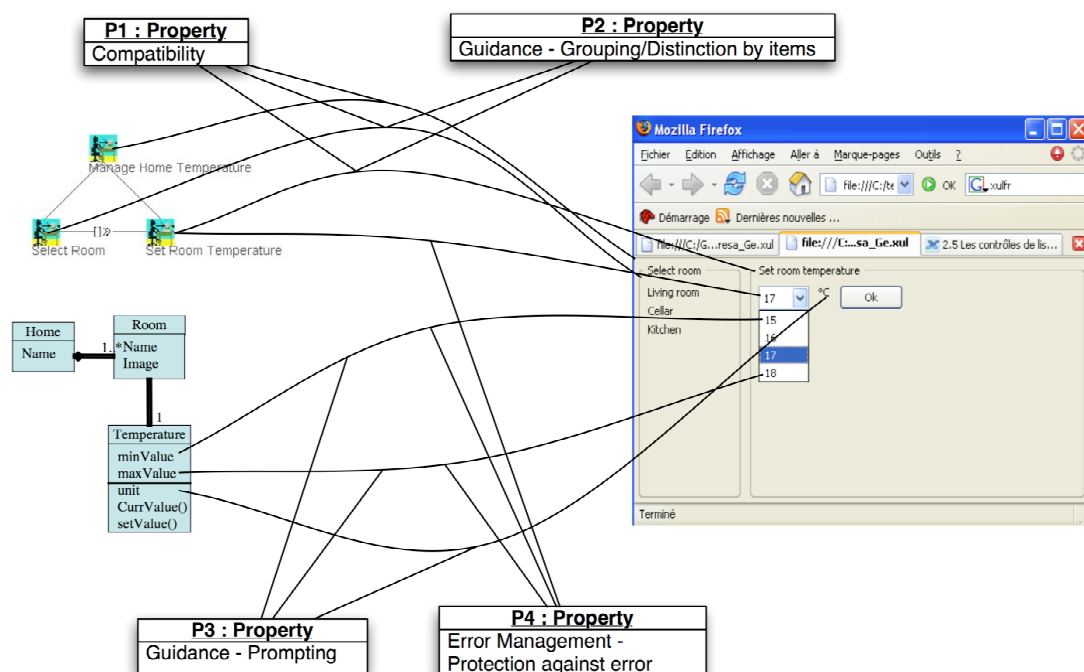


Figure 37 : Les transformations sont décrites en termes des propriétés qu'elles satisfont. Extrait de [Sottet 07b].

Principe #3 : Le graphe de modèles est complet ou incomplet

- Le système peut être généré à partir du graphe de modèles
- Le graphe de modèles peut être retrouvé à partir du système
- Le système et le graphe de modèles peuvent être établis puis mis en correspondance de façon complète ou partielle

L'approche est ouverte à toute pratique. Elle accueille les démarches :

- génératives descendantes (forward engineering) qui produisent le code à partir de modèles ;
- génératives ascendantes (reverse engineering) qui découvrent les modèles à partir de code ;
- déclaratives qui fournissent code, modèles et mises en correspondance.

Les modèles peuvent être partiels ; les démarches combinées. Ainsi, typiquement pour des techniques d'interaction post-WIMP, le programmeur branchera probablement des composants logiciels spécifiques alors que pour la structuration de l'IHM en espaces de dialogue il envisagera peut-être une génération à partir du modèle de tâches.

Principe #4 : Le graphe de modèles est présent à l'exécution et cohérent du système interactif

[Vanderdonckt 05] introduit la notion de *modèle actif* dans son espace problème. C'est cette notion qui est au cœur du principe #4. Le graphe de modèles est vivant à l'exécution et cohérent, dans son état, avec le système interactif :

- si l'état du système interactif change, la modification est propagée au modèle ;
- si le modèle change, le système interactif est mis à jour conformément.

La figure 38 présente un premier démonstrateur montrant le maintien de cohérence entre une interface concrète (à gauche) et l'interface finale qu'elle décrit (la fenêtre en bas à droite). L'interface finale est celle de HHCS pour deux pièces : le salon et la cuisine. L'interface concrète est le graphe de scènes de l'application. Il est affiché sous la forme d'un graphe d'ellipses blanches. Par une toolglass (fenêtre en bas à gauche), l'utilisateur peut supprimer des éléments de l'IHM concrète ou finale : il place la toolglass (carré vert en bas de la ToolGlass) sur l'élément à supprimer puis active le bouton de destruction. L'élément disparaît dans l'IHM manipulée. La modification est propagée à l'autre IHM :

- si l'utilisateur détruit le nœud « Kitchen » du graphe de scène, le lien hypertexte correspondant disparaît de l'IHM finale ;
- si l'utilisateur détruit le lien hypertexte « Living room » de l'IHM finale, le nœud correspondant disparaît du graphe de scène.

Sur ce démonstrateur, les modifications sont portées par l'utilisateur. La seule ambition ici était de montrer le maintien de cohérence entre système et modèle.

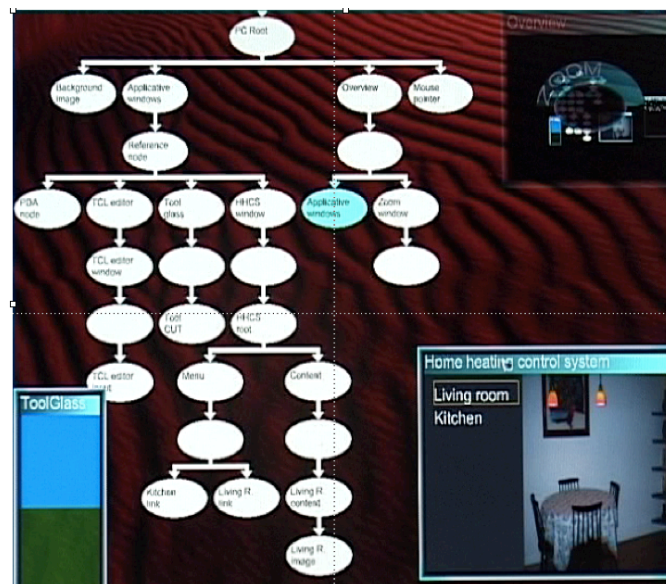


Figure 38 : Maintenance de cohérence entre système (IHM finale en bas à droite) et modèle (graphe de scène à gauche). Extrait de [Sottet 07d].

On appelle COSM (Cobweb of Models and Systems) le système formé du système interactif et de son graphe cohérent de modèles.

Principe #5 : Les modèles sont conformes à des métamodèles

La conformité à des métamodèles est fondamentale pour permettre au système de raisonner sur sa propre conception à l'exécution. Ceci concerne tous les modèles du graphe, y compris les transformations. La figure 39 donne un exemple de métamodèle de tâche. Ce métamodèle spécifie qu'une tâche (*Task*) se décompose en sous-tâches (*Task*) par le biais d'opérateurs binaires tels que les opérateurs Lotos (*BinaryOperator*). Une tâche décomposée est dite abstraite. Le métamodèle distingue les tâches abstraites de nature sémantique (*subTaskGrouping*) des tâches abstraites motivées par une factorisation soit d'opérateurs (*operatorFactorization*) soit de décorations (*decorationFactorization*). Une tâche peut être décorée d'opérateurs unaires exprimant l'optionnalité (*optional*), l'itération (*iteration*), la criticité (*criticity*) ou la fréquence (*frequency*).

Notre ambition n'est pas de définir un nouveau métamodèle de tâche mais d'illustrer, sur un exemple simple, cette notion de métamodèle.

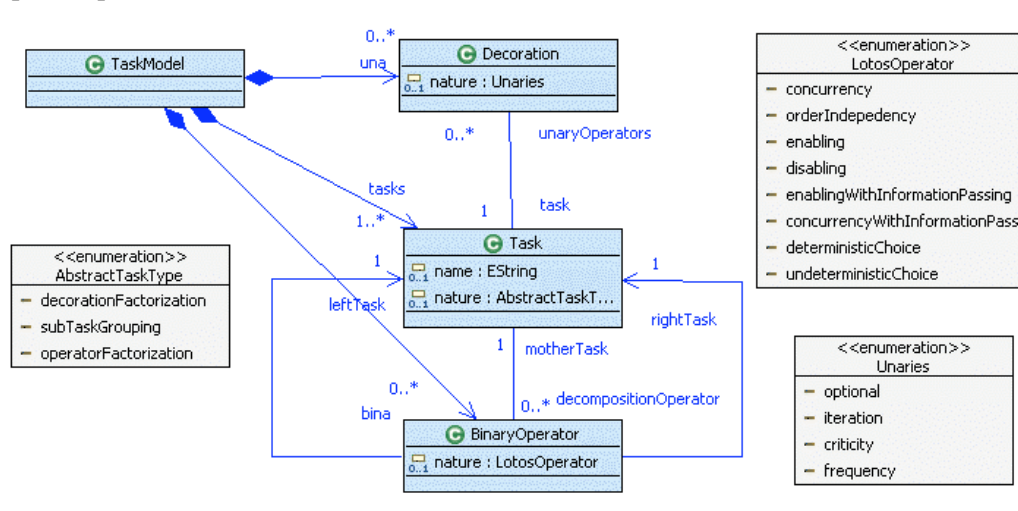


Figure 39 : Exemple de métamodèle de tâches. (JS Sottet).

[Sottet 07b] propose un métamodèle de mapping. Ce métamodèle intègre les propriétés d'ergonomie : c'est une originalité. Les propriétés sont exprimées dans un référentiel dont le choix est laissé au concepteur. Dans nos travaux, les autres métamodèles (tâches, concepts, IUA, IUC, plate-forme, utilisateur, etc.) sont rudimentaires à seule visée d'illustration de l'approche. Des langages tels qu'UsiXML en particulier font ici référence [url UsiXML].

La figure 40 montre un extrait du graphe de modèles pour le cas d'étude. Chaque modèle (*M1*) est conforme à un métamodèle (*M2*). Les liens sont établis entre métamodèles. Ils permettent, par exemple, d'exprimer que :

- une tâche (*Task*) manipule des concepts (*Concept*) ;
- une tâche donne lieu à des espaces de dialogue (*Workspace*) peuplés d'interacteurs (*Interactor*) ;
- les interacteurs concrétisent les concepts, tâches et espaces de dialogue, etc.

Pour des raisons de lisibilité, le métamodèle des mappings n'est pas présenté. Il en est de même pour le contexte d'usage, le noyau fonctionnel ainsi que les transformations de déploiement.

Selon la terminologie IDM [Favre 04d], la figure 40 est un extrait du mégamodèle pour l'IHM : il met en relation les modèles de l'IHM.

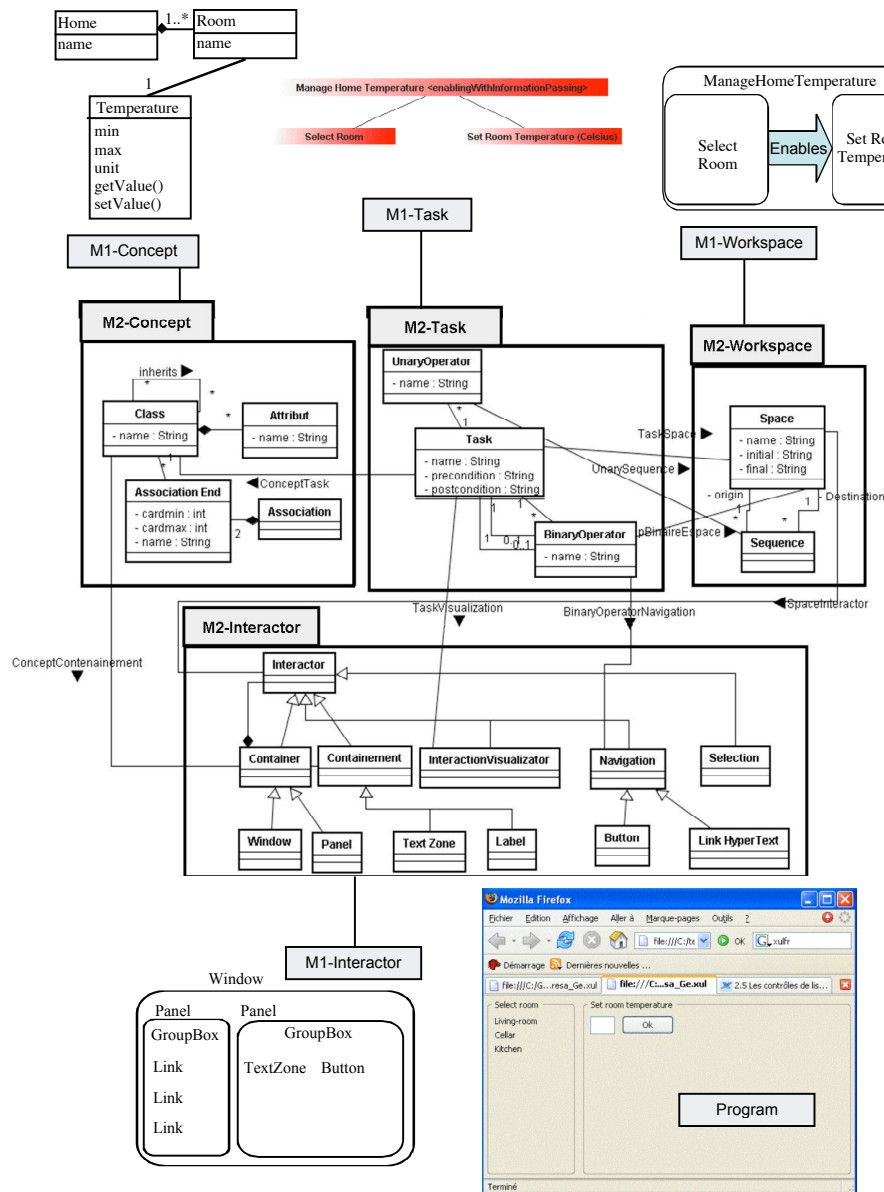


Figure 40 : Un extrait du mégamodèle pour l'IHM. Extrait de [Sottet 07b].

Principe #6 Le graphe de modèles est placé sous le contrôle de l'utilisateur

6a) Au niveau M1, l'IHM est dite Extra-IHM

6b) Au niveau M2, l'IHM est dite Méta-IHM

6c) Aux niveaux M1 et M2, l'IHM est dite Méga-IHM

Le graphe de modèles M1 et M2 est observable et/ou contrôlable par l'utilisateur. La toolglass de la figure 38 est un premier exemple d'Extra-IHM. Elle permet la manipulation de modèles du niveau M1. La figure 41 est un deuxième exemple d'Extra-IHM. Elle permet de remodeler et de redistribuer l'IHM par la manipulation de son modèle des tâches. Le modèle des tâches est affiché en partie gauche de la figure 41 sous la forme d'un arbre. En sélectionnant une tâche sur ce modèle, l'utilisateur (ici le concepteur faute d'une ergonomie travaillée) voit, en partie droite, les informations relatives à la tâche : son nom (*SelectRoom*) et son type (*Choix d'une pièce parmi N*). L'utilisateur peut changer le type de la tâche. Il peut, par exemple, la déclarer de type spécification libre : le nom de la pièce serait alors à saisir dans un champ texte. C'est un remodelage. Il peut aussi modifier l'allocation des tâches et

sous-tâches aux plates-formes connectées. Les plates-formes sont affichées en bas à droite de la fenêtre. Sur la figure 41, deux plates-formes sont connectées : un PDA HTML et un PC XUL. En sélectionnant une tâche (en partie gauche) puis en cochant les plates-formes sur lesquelles elle doit apparaître (en bas de la partie droite), la redistribution de l'IHM s'opère conformément à cette spécification.

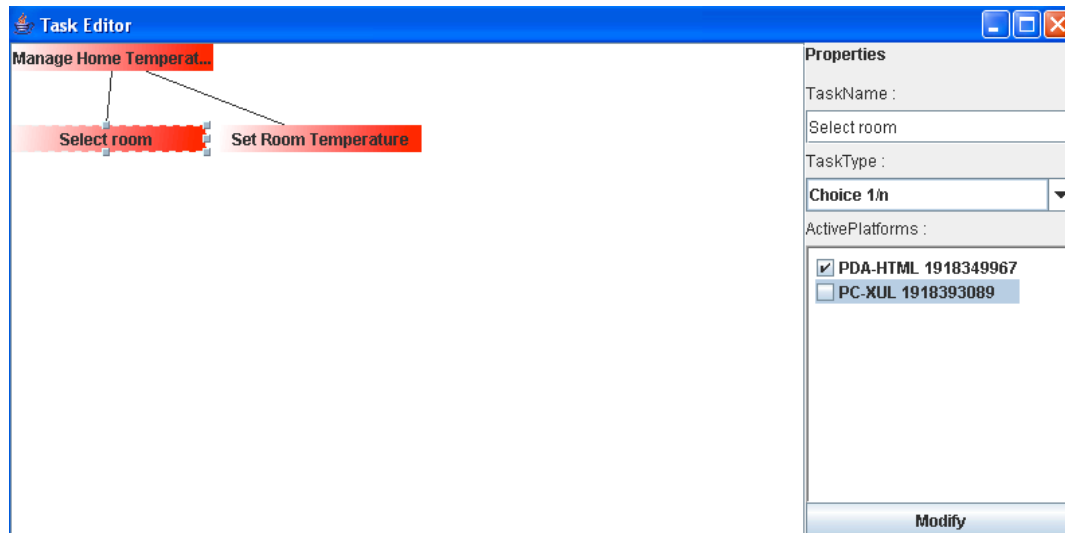
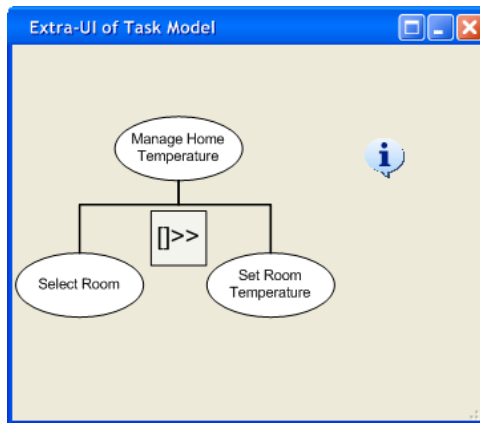


Figure 41 : Modèles de tâches et de plates-formes observables pour le contrôle par l'utilisateur des tâches et de leur distribution. (X. Alvaro et JS Sottet).

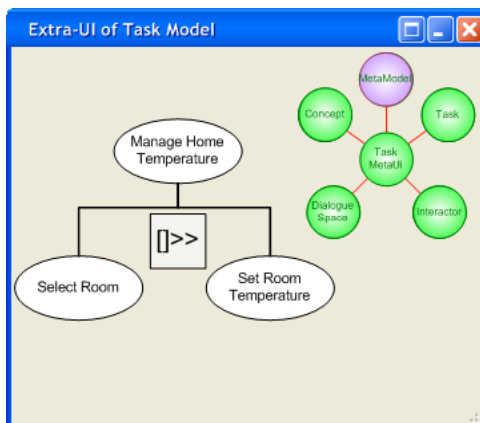
La figure 42 présente quelques maquettes non implémentées marquant la différence entre Extra- et Méta-IHM. Le modèle considéré est le modèle des tâches. L'Extra-IHM place le modèle des tâches sous le contrôle de l'utilisateur. La présentation adoptée est un arbre. Un point d'information « i » (Figure 42a) permet d'accéder aux autres modèles du graphe de modèles : l'accès à ces modèles est donné par un menu en fleur (Figure 42b). Le menu donne accès aux autres modèles M1 (concepts, espaces de dialogue et interacteurs) mais aussi au métamodèle de niveau supérieur : le métamodèle de tâches. La sélection de cette option (métamodèle) donne accès à une Méta-IHM qui rend observable et contrôlable le métamodèle de tâches (Figure 42c) : elle permet de modifier le langage de définition des tâches. Par exemple, si l'utilisateur souhaite rendre les opérateurs entre tâches N-aires au lieu de binaires, c'est à ce niveau qu'il convient d'agir. La Méta-IHM rejoint le Meta-Design de [Fischer 04] : l'utilisateur n'est plus bridé comme simple exécutant. Il prend part à la conception du système interactif par la mise à disposition de systèmes ouverts, volontairement sous-conçus (« underdesigned »).

Les Extra-, Méta- et Méga-IHM sont des concepts nouveaux qui requièrent une étude ergonomique. Elles justifieront peut-être la création d'interacteurs dédiés.

(a) Extra-IHM pour l'observabilité du modèle de tâche et l'accès aux autres modèles



(b) Extra-IHM de a) après activation du « i »



(c) Meta-IHM de tâches suite à la sélection de MetaModel en b)

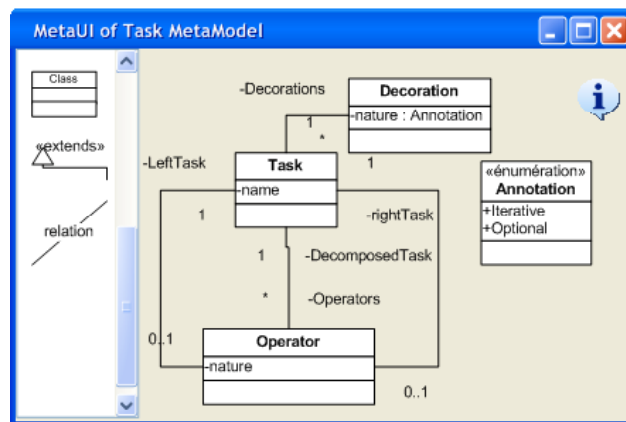


Figure 42 : Maquettes non implémentées d'Extra- et Méta-IHM. (JS Sottet).

Principe #7 La plasticité est une transformation

La plasticité est une transformation du COSM sur changement des modèles du contexte d'usage (utilisateur, plate-forme et/ou environnement). Elle est dictée par un modèle de valeur et d'évolution.

Le modèle de valeur spécifie la valeur du système interactif attendue. Le modèle d'évolution est un graphe de contextes "clé" (Figure 43), c'est-à-dire de contextes singuliers ayant été jugés pertinents par le concepteur et/ou ayant été appris automatiquement ou sous le contrôle de l'utilisateur final. Les arcs sont les changements de contexte d'usage. Des transformations leur sont attachées.

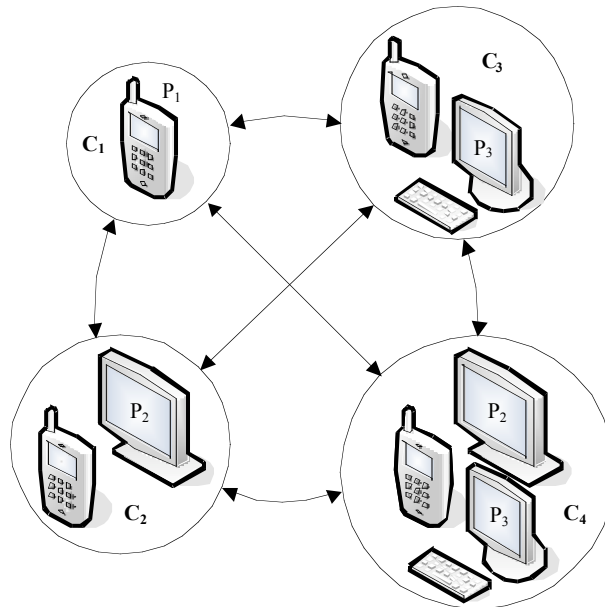


Figure 43 : Contextes d'usage clé (C1 à C4) et changements de contexte d'usage. (V. Ganneau).

Principe #8 Les systèmes sont capitalisés

La capitalisation du savoir-faire est un élément essentiel. Elle doit agir aux niveaux M1 et M2. Le niveau M1 est pertinent pour réutiliser des systèmes existants. Le niveau M2 est indispensable dès lors que le système doit raisonner sur le niveau M1. La capitalisation est au cœur du projet ZOOOMM¹² (ZOO Of MetaModels) [url ZOOOMM].

Principe #9 Les outils de développement d'IHM sont des transformateurs

Les modèles de transformation sont exécutés par des transformateurs. Les outils de développement d'IHM (éditeurs, générateurs) jusqu'ici cantonnés à la phase de conception sont des transformateurs d'IHM. Ils sont invocables à l'exécution. Les algorithmes d'Intelligence Artificielle pour la perception du contexte d'usage, l'apprentissage, etc. sont d'autres exemples de transformateurs. Les transformateurs agissent sur le système ou son graphe de modèles. La cohérence est assurée d'après le Principe #4.

Principe #10 Les modèles de transformations et transformateurs sont internes ou externes

D'un point de vue de l'architecture logicielle, l'adaptation est réalisée de manière interne ou externe à l'IHM. Dans le cas d'une adaptation externe, c'est un autre système (système interactif ou infrastructure d'exécution) qui la prend en charge.

Les principes de l'approche étant énoncés, nous les examinons d'un point de vue pratique dans la section suivante.

¹² L'initiative remonte aux années 2000. Des centaines de métamodèles sont aujourd'hui recensés. Le recueil se fait par l'analyse d'articles scientifiques et l'organisation de safaris dans diverses universités. Cf Jean-Marie Favre.

Chapitre 3. Des axes de recherche pour construire



La plasticité, un sujet en trois temps.

Troisième mesure : donner corps à l'approche scientifique en des axes de recherche.

L'approche étant fixée, ce chapitre décrit, en pratique, l'organisation du travail. Des axes de recherche sont définis : ils constituent chacun un sujet de thèse. Nous distinguons les axes de travail actuels (section 1) des travaux futurs (section 2).

1. Axes de travail actuels

Nos efforts se sont aujourd'hui portés sur l'appropriation de l'IDM pour l'IHM ; l'étude d'interacteurs pour la plasticité et la compréhension des directives d'adaptation. Une infrastructure logicielle a été développée pour intégrer ces travaux. Nous les présentons dans cet ordre.

1.1 Ingénierie Dirigée par les Modèles

Etudiant : Jean-Sébastien Sottet.

Thèse : 2005 à 2008.

Co-encadrement avec : Jean-Marie Favre, laboratoire LIG, équipe ADELE.

Financement : Bourse MENRT.

Projets : IMAG MAPPING et européen ITEA EMODE.



En 2003, sans nous placer sous l'égide de l'IDM, nous proposons un cadre de référence pour la conception et l'exécution d'IHM plastiques (Figure 44) [Calvary 03]. Ce cadre de référence unifiait la conception et l'exécution autour de trois types de modèles : les modèles ontologiques, archétypes et observés. Ces modèles correspondent en fait aux trois niveaux de la pyramide de l'OMG : les métamodèles (M2), les modèles (M1) et la trace à l'exécution (M0). Outre l'identification des métamodèles, ce cadre de référence identifiait des chemins de conception. Il rendait explicite la nécessaire conservation de l'information au fil de l'ingénierie. Par exemple, dans une démarche descendante, l'interface abstraite (modèle D9) doit connaître les modèles de concepts et de tâches (D8) ayant présidé à sa conception.

Pour l'exécution, le cadre de référence intégrait les trois fonctions clé de l'adaptation : la détection du contexte d'usage (S pour Sensing) ; le calcul de la réaction (C pour Computing) et l'exécution de la réaction (E pour Executing). Il rendait explicites les deux allocations possibles de ces mécanismes : dans l'interface finale (interne) ou une infrastructure d'exécution (externe).

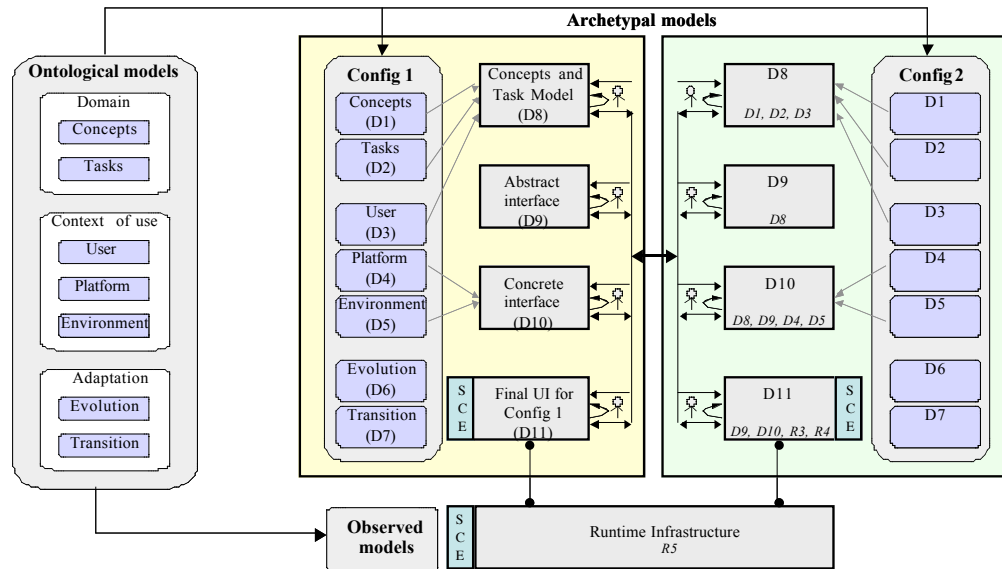


Figure 44 : Le cadre de référence unifié. Extrait de [Calvary 03].

Depuis, grâce au rapprochement avec l'IDM, nous avons reformulé et étendu ce cadre de référence (Figure 45) :

- Les niveaux M0, M1 et M2 sont explicites ;
- La modélisation est étendue aux scénarii d'usage et propriétés ;
- Les modèles sont conformes à des métamodèles ;
- Les transformations sont explicites couvrant, à la fois, les heuristiques de construction et d'adaptation d'IHM.

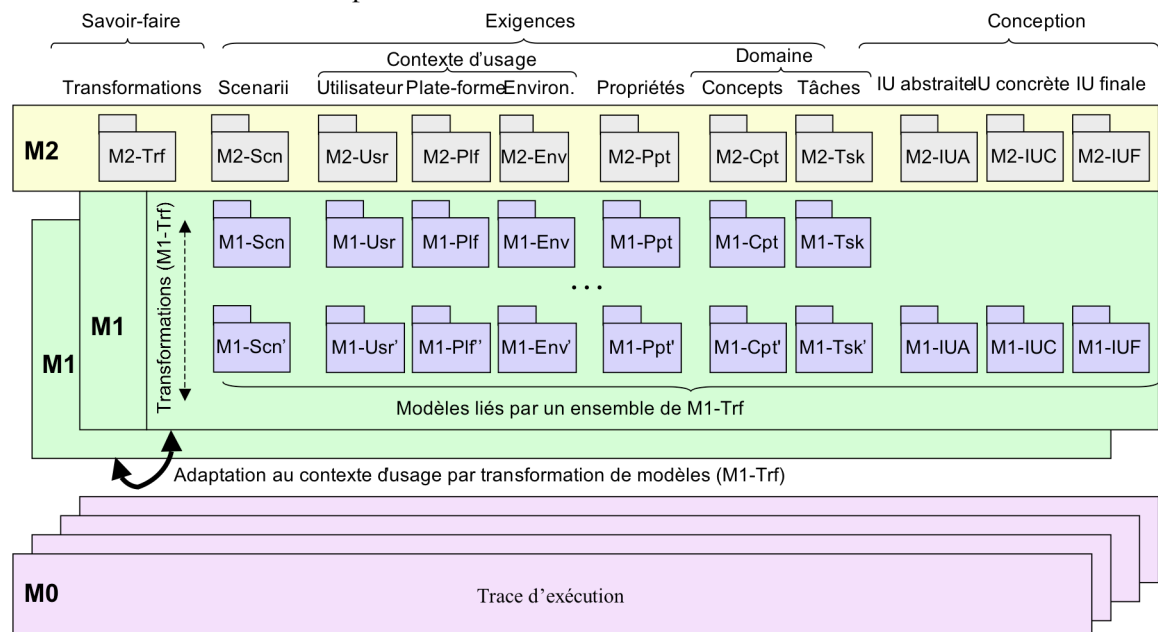


Figure 45 : Facture IDM du cadre de référence. Version étendue de [Sottet 06].

En pratique, un effort a été porté sur la maîtrise des concepts et outils de l'IDM. Par rapport au cadre de référence, les scénarii n'ont pas été traités. Des métamodèles ont été définis pour les autres modèles. D'un point de vue méthodologique, les transformations de conception sont descendantes : l'IHM est générée à partir de modèles.

D'un point de vue outils, les langages ciblés sont d'utilité publique : XUL et HTML. Les transformations sont des règles ATL. Elles couvrent à la fois les transformations verticales de génération d'IHM et horizontales de passerelles entre technologies. [Sottet 07c] montre un exemple de règle ATL permettant le passage de nos métamodèles à UsiXML pour l'invocation ensuite de l'arsenal d'outils UsiXML. Nos métamodèles sont écrits en EMF.

L'objectif est à terme de couvrir d'autres démarches et outils pour apprendre à la fois méthodologiquement et techniquement. En particulier, nous souhaitons, dans la génération d'IHM, cibler des boîtes à outils spécifiquement façonnées pour la plasticité : les COMET. Elles font l'objet de la section suivante.

1.2 Boîte à outils d'interacteurs pour la plasticité

Etudiante : Olfa Dâassi.
Thèse soutenue le 30 janvier 2007.
Financement : Projet européen IST CAMELEON.
Projet : CAMELEON.



La thèse propose une nouvelle génération d'interacteurs, les COMET, pour la construction d'IHM plastiques. Un interacteur est défini comme une entité numérique offrant des fonctions de service rendues observables et/ou manipulables à l'utilisateur par le biais d'entités physiques qui jouent le rôle de ressources d'interaction. Une COMET (COntext Mouldable widGET) est un interacteur façonné pour l'adaptation : il est doué d'autodescription. Il peut être polymorphe et/ou doué de capacité d'adaptation.

Alexandre DEMEURE étoffe le concept d'un point de vue conceptuel et implémentatif pour, en particulier, l'ouvrir à des contextes d'usage non prévus à la conception et à des incarnations multi-technologies.

Etudiant : Alexandre Demeure.
Thèse soutenue le 11 octobre 2007.
Financement : Bourse Présidence UJF.
Projets : IMAG MAPPING et européen IST CAMELEON.



La thèse étoffe les COMET d'un style d'architecture logicielle appelé COMET. Ce style fait le pont entre deux des approches phare en adaptation des IHM : les approches IDM et les approches par boîtes à outils. Il est régi par deux grands principes :

- Premièrement, un interacteur COMET est un système interactif. On doit donc pouvoir le considérer selon les différents points de vue classiques en IHM : le noyau fonctionnel, d'une part, et les tâches, interfaces abstraite, concrète et finale pour l'IHM, d'autre part ;
- Deuxièmement, les COMET ne visent pas à remplacer les Bào d'interacteurs existantes. Elles doivent, au contraire, en tirer profit, c'est-à-dire s'appuyer sur ces Bào pour les mises en œuvre technologiques.

En terme de vocabulaire, le style COMET définit trois éléments clé, correspondant chacun à une préoccupation particulière :

- Le LC (pour Logical Consistency) est l'élément central de la COMET : sa « raison d'être ». Il porte la sémantique du « service » rendu. Il peut être de deux types : soit une tâche utilisateur (par exemple, spécifier, s'identifier, etc.), soit un opérateur entre tâches (entrelacer, séquencer, etc.). La sémantique se manifeste, en particulier, par une API dite « sémantique ». Elle se décline en différents modèles logiques (LM) dont le LC a la charge de maintenir la cohérence ;
- Un LM (pour Logical Model) est chargé d'une préoccupation particulière dans la réalisation du LC. Les exemples classiques de LM sont les LM de présentation chargés de présenter la tâche à l'utilisateur et les LM fonctionnels chargés de la mise en œuvre algorithmique du LC indépendamment de toute présentation. Chaque LM doit implémenter l'API sémantique du LC : c'est le langage qu'ils partagent. Le LM peut éventuellement étendre cette interface pour prendre en compte des considérations qui lui sont propres. Le LM est chargé de maintenir une cohérence entre les différents modèles physiques (PM) qui lui sont liés. Il fournit de plus des usines à PM pour l'instanciation de nouveaux PM ;
- Un PM (pour Physical Model) est une façon particulière de réaliser un LM. Un PM encapsule du code de BàO. Par exemple, un PM de présentation associé à la COMET de Log peut encapsuler le code correspondant à des champs textes de la BàO HTML ou TK, SWING, etc. Le code de la BàO utilisé dans le PM est appelé « primitive technologique ». Le PM doit implémenter l'API sémantique du LM qui lui est associé : c'est leur « langage » commun. Il fournit de plus une description du contexte d'usage qui lui est nécessaire. Par exemple, il exprime le fait qu'il lui faille une plateforme matérielle de type PC et logicielle de type Java 1.4.

Du point de vue des contraintes de configuration, une COMET est définie par un LC. Un LC est connecté à un ou plusieurs LM, eux-mêmes connectés à un ou plusieurs PM. LC et LM d'une part, LM et PM d'autre part communiquent bidirectionnellement. La figure 46 propose une représentation graphique pour les COMET. Elle est appliquée à la COMET de Log.

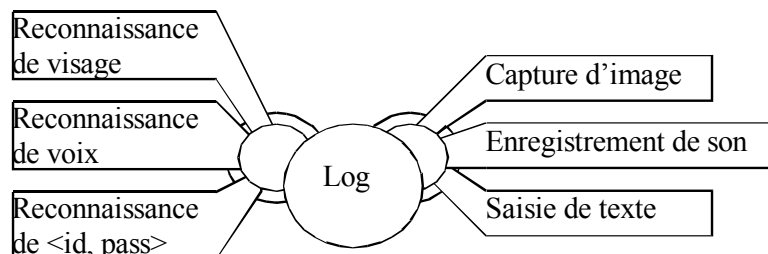


Figure 46 : Représentation graphique des COMET appliquée à la COMET de Log : le LC est au centre. Il communique ici avec deux LM, dotés chacun de trois PM. Extrait de [Demeure 07].

Dans la COMET, le LC gère la compatibilité entre les LM d'abstraction et de présentation ainsi que la cohérence de la COMET. Prenons l'exemple du Log. S'il n'existe pas de PM fonctionnel permettant une analyse de visage, alors le LC proscrira au LM de présentation l'utilisation d'un PM de ce type (compatibilité). Si l'une ou l'autre des facettes est modifiée, la modification est propagée aux autres facettes (cohérence). Supposons que l'API sémantique du Log incluent les méthodes *reset()* et *Log()*. La méthode *reset()* peut être déclenchée directement au niveau du LC (Figure 47-A). Le LC propage l'appel à ses LM qui le propagent à leurs PM. La méthode *log()* peut être déclenchée à partir d'un PM, par exemple, d'un PM de saisie de couple <identifiant, mot de passe>. Le PM propage l'appel à son LM (Figure 47-B). Celui-ci détermine dans quelle mesure l'appel doit être propagé aux autres PM. Il peut y avoir différentes politiques à ce niveau. L'utilisateur doit-il se logger sur chaque PM ou bien un seul suffit-il ? Une fois cela déterminé, le LM transmet l'appel à son LC qui le transmet aux autres LM. Ces LM le transmettent à leur tour à leurs PM.

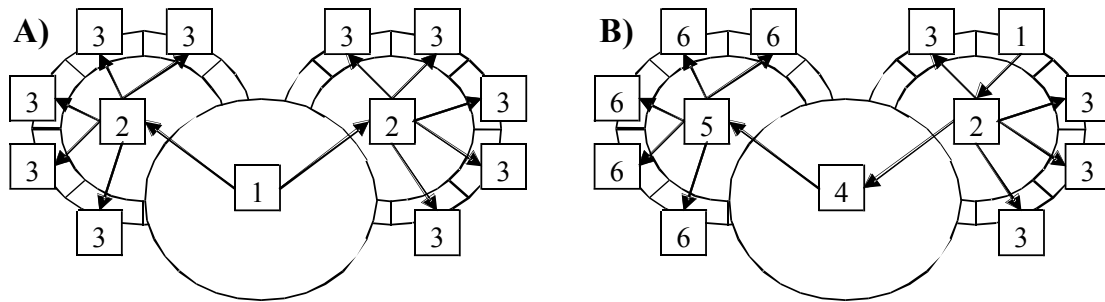


Figure 47 : Propagation des appels de méthode pour la cohérence de la COMET. Les numéros représentent l'ordre d'appel. Les flèches indiquent le sens de propagation. Extrait de [Demeure 07].

Les COMET s'assemblent en graphes orientés. Elles donnent lieu à trois graphes parallèles : le graphe des LC, le graphe des LM de présentation et le graphe des PM de présentation. L'orientation des graphes correspond à la relation « père-fils » c'est-à-dire « s'exprime dans ». Par exemple, pour les PM de présentation graphique, « PMF est fils de PMP » signifie que l'affichage de PMF dépend de PMP. La figure 48-B) reprend l'exemple du Log mais enrichi d'une image et d'un texte de publicité. Le graphe sémantique des COMET (graphe des LC) est donné en A) : il y a entrelacement entre le log et la consultation de l'image et du texte. Image et texte sont des COMET.

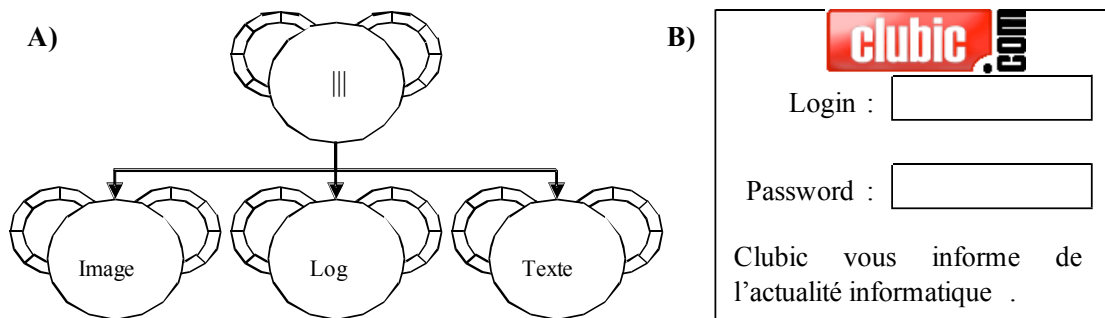


Figure 48 : A) Entrelacement de trois COMET : une image (titre), un log et un texte (explicatif). B) Un rendu possible. Extrait de [Demeure 07].

Les graphes sont hiérarchiques : chaque élément peut contenir des COMET. L'élément est alors dit « composite ». On peut, dans ce cas, spécifier des points d'entrée et de sortie entre l'élément composite et son graphe d'appartenance. La figure 49 donne un exemple pour un PM de la COMET Log construit par insertion d'un identifiant et d'un mot de passe dans une fenêtre de confirmation. Le point d'entrée est un PM de la COMET Validation.

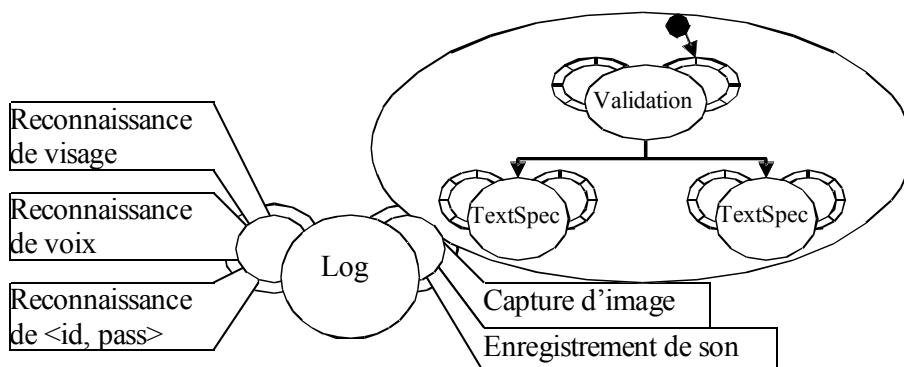


Figure 49 : Exemple de graphe composite pour la COMET Log.

Les COMET peuvent être rendues simultanément et dans plusieurs technologies : PM TK, AJAX, B207 (post-WIMP) et S207 (vocal). Elles sont extensibles dynamiquement grâce à un graphe des descriptions (GDD) jouant le rôle d’annuaire d’IHM. Les LM gèrent la multimodalité (équivalence et redondance). Un langage COMET/RE est défini à ces fins. Les COMET sont transformables par un langage de style CSS++.

1.3 Directives d’adaptation

Etudiant : Vincent Ganneau.

Thèse : 2005-2008.

Co-encadrement avec : Rachel Demumieux, ergonome à France Telecom.

Financement : Contrat France Telecom Lannion.



La thèse se situe à la frontière du Génie Logiciel et de l’Intelligence Artificielle. Elle marie l’IDM et l’apprentissage pour la capitalisation des directives de valeur. [Ganneau 07] propose un métamodèle de directive d’adaptation. Ce métamodèle est à visée descriptive et générative [Beaudoin-Lafon 00]. Il s’agit, d’un point de vue conceptuel, d’aider le concepteur à se poser les bonnes questions en matière de plasticité et de l’accompagner, d’un point de vue implémentationnel, dans la mise en œuvre de l’adaptation. La proposition est un outil partenaire instrumentant le code du système interactif conformément aux directives d’adaptation : la partie gauche des directives définit les points de l’écosystème à sonder. Toute variation dans l’un des points sondés est transmise au moteur d’adaptation (① et ② sur la figure 50). Le moteur d’adaptation est en charge de choisir la réaction appropriée. Le choix s’appuie, d’une part, sur les directives d’adaptation et, d’autre part, sur un modèle de l’utilisateur mis à jour dynamiquement au fil de l’interaction. L’adaptation est placée sous le contrôle de l’utilisateur via une Extra-IHM. L’Extra-IHM permet à l’utilisateur d’avaliser ou non (④) les propositions système d’adaptation (③). Les décisions de l’utilisateur alimentent une gestion des préférences dans le modèle de l’utilisateur.

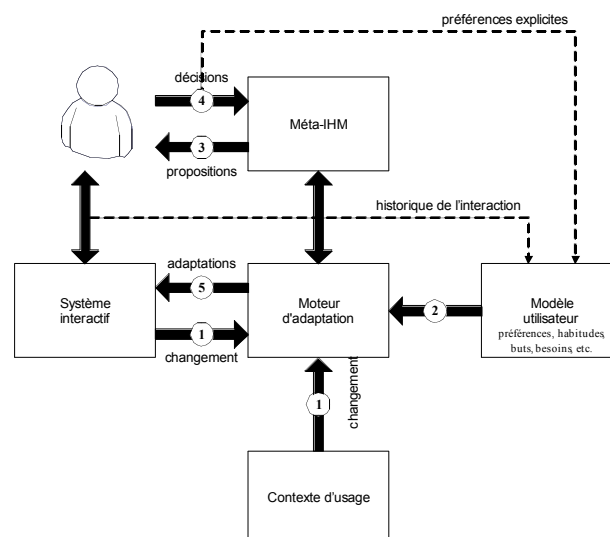


Figure 50 : Décomposition fonctionnelle de l’adaptation. Extrait de [Ganneau 07].

Les modèles utilisateurs ont été particulièrement exploités dans les systèmes d’aide intelligents [Brusilovsky 97] et les interfaces adaptatives [Liu 03]. Le modèle utilisateur contient toute la connaissance que le système a de l’utilisateur. Les attributs typiquement maintenus dans le modèle sont les préférences, intérêts, buts et besoins d’un utilisateur ainsi que l’historique de l’interaction (tâches en cours, tâches terminées, erreurs, etc.). L’historique est aujourd’hui gérée dans l’infrastructure logicielle MARA (Models At Runtime for Adaptation) grâce aux transformations qui immiscent dans le code du système interactif les

informations nécessaires à cette gestion : chaque interacteur sait à quelle tâche il correspond. Cette infrastructure fait l'objet de la section suivante.

1.4 Infrastructure logicielle pour la capitalisation

MARA (Models At Runtime for Adaptation) est une infrastructure logicielle pour l'intégration de nos travaux. Elle a été développée par Xavier Alvaro, ingénieur financé sur le projet européen ITEA EMODE. Tous nos travaux en plasticité y prennent place, à l'exception des COMET non encore intégrées aujourd'hui.

L'infrastructure est conforme à la décomposition fonctionnelle de [Balme 04]. Elle suit, d'un point de vue implémentatif, une approche à services. Elle est implémentée en OSGi. Nous l'illustrons sur le cas d'étude HHCS (chapitre 2). Imaginons qu'un usager veuille contrôler la température de sa maison à partir d'un navigateur Internet. Ce navigateur est un client Ajax (bloc gauche de la Figure 51). Il se connecte via un serveur http à une servlet « Emode », dédiée à la gestion du confort. Cette connexion déclenche la mise à jour du modèle de contexte dans le service *EcosystemManager* (partie haute du bloc droit de la Figure 51) en charge du graphe de modèles : la plate-forme de l'utilisateur (un PC ou PDA HTML ou XUL) est ajoutée au modèle de plate-forme. Le contexte d'usage ayant changé, l'IHM doit être (re)calculée : c'est le rôle du service *UI Transformer*. Ce service est une machine virtuelle ATL. Il consulte le moteur d'adaptation (*EvolutionEngine*) pour connaître la directive à appliquer. Les directives sont gérées dans un service (*TransformationRulesManager*). Ce sont des règles ATL. Elles manipulent des métamodèles d'entrée et de sortie, gérés dans un service dédié : le *MetaModelsManager*. Ces métamodèles sont exprimés en EMF.

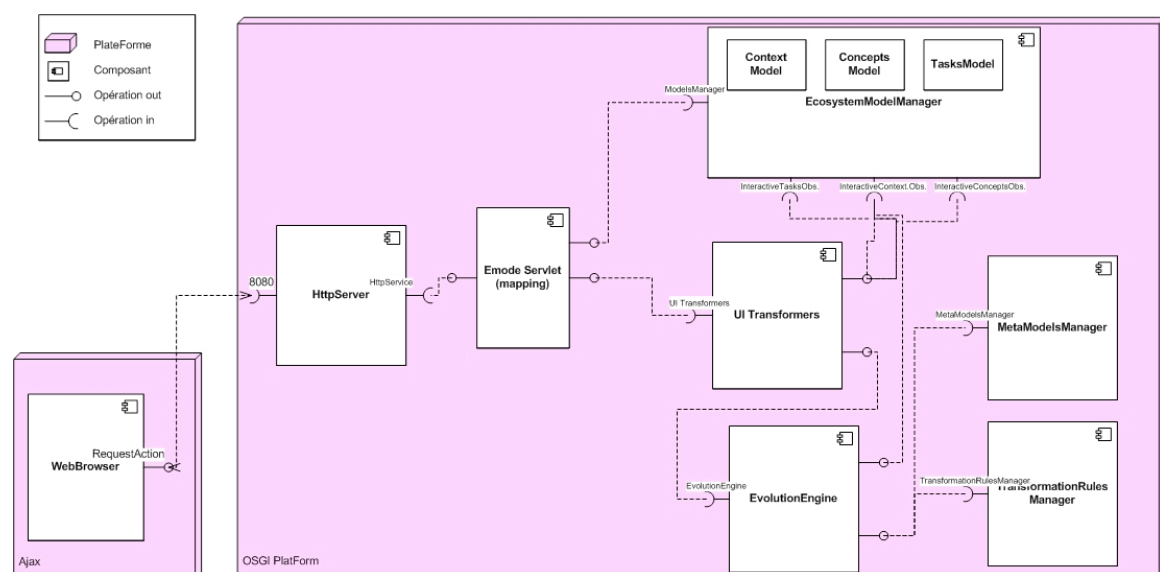


Figure 51 : Infrastructure logicielle MARA. (Xavier Alvaro).

L'infrastructure se veut fédératrice sur le long terme. Elle accueillera les nouveaux travaux.

2. Axes de recherche naissants

Nous présentons ici les sujets dans un ordre de maturité décroissant.

2.1 Composition d'IHM par Planification Multi-Agent

Etudiant : Yoann Gabillon.

Thèse : 2007-2010.

Co-encadrement avec : Humbert Fiorino, laboratoire LIG, équipe MAGMA.

Financement : Bourse Rhône-Alpes, cluster ISLE, projet PRESENCE.

Le projet IMAG MAPPING a exploré la pertinence de la planification multi-agent pour la plasticité. Rémi Dupuis (RICM2, 2005-2006) puis Vincent Paya (M2R, 2006-2007) se sont relayés sur le sujet.

En Intelligence Artificielle, la planification multi-agent¹³ sous hypothèses propose des mécanismes de synchronisation/coordination de plans qui ont été produits localement par des agents poursuivant des objectifs différents. L'enjeu est double : d'une part, éviter les conflits d'accès aux ressources, d'autre part, augmenter les performances (par exemple, en détectant des activités redondantes). Ici, l'élément déterminant n'est plus la gestion efficiente des conflits mais la potentialisation des compétences : les agents co-construisent une solution globale. Nous identifions deux grandes classes de scénarii opérationnels :

- La planification au service de la plasticité : il s'agit, par la planification multi-agent, de construire une IHM conforme au besoin de l'utilisateur et compatible du contexte d'usage courant. Dans notre premier prototype (Rémi Dupuis), l'utilisateur spécifie son objectif en langage naturel dans une fenêtre dédiée : par exemple, assister à des conférences traitant de la plasticité. La phrase est analysée par ConceptNet, une base de données et un moteur de bon sens. Les agents co-construisent alors un plan tenant compte de leurs compétences (c'est-à-dire leur noyau fonctionnel) et de leurs ressources respectives : quelle plate-forme prend en charge quelle sous-tâche et selon quelle modalité, sachant que (a) chaque agent dispose d'une représentation locale et par conséquent partielle voire contradictoire du monde et des compétences pouvant être mises en œuvre par ailleurs ; (b) ces compétences ne sont pas stables (pannes, ajout ou retrait de plates-formes par l'utilisateur, ressources consommables) et (c) le plan courant doit être révisé pour garantir dans la mesure du possible l'intégrité fonctionnelle du système ;
- La plasticité au service de la composition des compétences : la co-construction de plan peut aussi être envisagée au niveau fonctionnel. Dans ce cas, on cherche à combiner les compétences de chaque agent afin d'obtenir un résultat qu'aucun agent n'aurait pu atteindre seul : par exemple, une personne arrivant dans une gare demande à son PDA (équipé d'une carte wifi et d'un récepteur GPS) de lui trouver un hôtel proche du centre de conférences. Le PDA entre en relation avec les agents "transports urbains", "service de réservation" et "accueil des voyageurs" afin de produire le plan satisfaisant son but : "Attente taxi longue. Possibilité de prendre le métro jusqu'à la station... Se rendre à l'hôtel... par la rue..." etc. Il faut alors élaborer une IHM pour l'observabilité et la réalisation éventuelle du plan.

Dans nos premiers travaux, la construction de l'IHM suit une approche IDM. Elle est dirigée par l'objectif que l'utilisateur spécifie. Elle s'appuie sur la capitalisation d'IHM élémentaires (des microCOSM). Ces IHM sont assemblées pour tenir l'objectif utilisateur spécifié. Aujourd'hui, la chaîne IHM-Planification-IHM est couverte en largeur. Les données d'entrée sont les suivantes :

¹³ Extrait de la proposition MAPPING. Partie écrite avec Humbert Fiorino.

- L'objectif utilisateur est une tâche utilisateur sans procédure. Elle est conforme au métamodèle de JS Sottet ;
- Les IHM élémentaires sont des modèles de tâches conformes au métamodèle de tâches. Elles rendent explicites leur requis en terme de modèle de plate-forme (ressources d'interaction d'entrée et de sortie) ;
- Un état du monde est déclaré sous la forme d'un modèle de plates-formes.

Ces données d'entrée sont traduites par une règle ATL dans le domaine de la planification. Une correspondance partielle est aujourd'hui faite entre les opérateurs de tâches en IHM et le domaine de la planification. L'algorithme de planification est JSHOP. Il calcule un ensemble de solutions dans le domaine de la planification. Ces solutions sont ensuite transcrites en modèles de tâches par une transformation ATL. Ces modèles sont affichés (un par onglet) dans une IHM. La composition d'IHM concrètes [Lepreux 06] n'est pas traitée ; l'Extra-IHM non plus.

Dans le sujet, nous identifions deux dominances :

- IHM, d'une part, avec (1) l'étude de la description à donner aux IHM élémentaires en termes de ressources d'interaction requises et qualité ergonomique satisfaite, (2) la définition de patrons de composition d'IHM et (3) l'identification de points de contrôle de l'ergonomie dans le processus de composition d'IHM par planification multi-agent ;
- Planification, d'autre part, avec la difficulté de l'exigence « temps-réel ». Les algorithmes « any time » seront, en particulier, à étudier. Le grain et la distribution des agents seront aussi à étudier.

L'Extra-IHM ne fait pas partie du sujet. Elle fait l'objet d'un sujet dédié.

2.2 Méga-IHM

Sujet de M2R ouvert en 2007-2008.
Co-encadrement avec : Joëlle Coutaz.

Si l'autonomie des systèmes est un enjeu important de l'informatique ambiante, il convient néanmoins que les choix d'adaptation effectués par ces systèmes soient conformes aux attentes de l'utilisateur. Or, les intentions humaines ne sont pas toujours identifiables. De même, la technologie est souvent sujette à panne. L'objectif de l'étude est de transformer ces faiblesses techniques en réintroduisant l'utilisateur dans la boucle d'autonomie. Il s'agira d'inventer et de réaliser une suite d'outils permettant à l'utilisateur non-spécialiste en informatique de programmer, de mettre au point, voire de réutiliser des programmes lui permettant d'observer, de tracer et de contrôler l'autonomie des systèmes ambiants. L'enjeu est que ces activités de développement soient adaptées à un non-expert. Elles couvrent les niveaux M1 (Extra-IHM) et M2 (Méta-IHM).

2.3 Variabilité de la Valeur

Sujet de M2R ouvert en 2007-2008.
Co-encadrement avec : Jean-Sébastien Sottet et Sophie Dupuis.

La conception d'une IHM suit un ensemble de compromis, sacrifiant ou favorisant certains critères. En conséquence, l'IHM résultante ne peut être vue comme le reflet fidèle du cahier des charges. Elle ne peut donc pas servir de valeur de référence à l'adaptation. Le sujet proposé traite de la variabilité des spécifications à la conception et à l'exécution. Il s'agit de garder trace des transformations pour mesurer la conformité de l'IHM aux spécifications.

Nous reprenons la métaphore des élastiques [Calvary 98] : des liens sont tissés entre les spécifications, d'une part, et l'IHM, d'autre part. La tension des élastiques reflète les limites de plasticité. Le sujet requiert la définition de critères et métriques pour la valeur. Ceci sera étudié avec Dominique Scapin dans le projet ANR 2007-2010 MyCitizSpace.

Conclusion et perspectives : deux temps pour l'avenir



*La plasticité, un sujet en trois temps.
Deux mesures en réserve pour les perspectives.*

Ce document relate sept années de recherche sur la plasticité. « Sept ans : l'âge de raison » dit-on. Nous avons le sentiment aujourd'hui d'avoir creusé, creusé et encore creusé pour bien comprendre l'ingénierie d'IHM traditionnelles et l'espace problème de la plasticité : c'est cette compréhension que nous souhaitons, en premier lieu, transmettre dans ce document.

Nous avons proposé une approche dont l'originalité est double : d'une part, la mixité des « techniques » (GL dans ses démarches ascendante et descendante, ses approches à services, etc. ; IA par l'apprentissage et la planification multi-agent) ; d'autre part, l'abandon d'une dichotomie entre conception et exécution, dichotomie perpétuée par héritage. Cette dichotomie aujourd'hui n'a plus de sens : nous la brisons et nous ouvrons, en conséquence, sur le end-user programming.

Si, sur ce point, nous rompons avec le passé, nous le respectons par ailleurs. En particulier, nous espérons, par l'approche IDM, réhabiliter les modèles : ils sont désormais conformes à des métamodèles et vivants à l'exécution pour que le système puisse raisonner dynamiquement sur sa propre conception. [Myers 00] laissait percevoir cet espoir pour les approches à base de modèles : elles avaient été cisailées par les tentatives décevantes de génération d'IHM. Toutefois, d'un point de vue méthodologique, les modèles ont fait leur preuve en IHM : notre approche souhaitait profiter de ce savoir-faire. Nous ressuscitons les modèles dans la sécurité grâce au principe directeur de mixité.

Par l'approche IDM, nous mettons aussi au jour un concept intégrateur : la Méga-IHM. Par ce concept, nous avons le sentiment d'avoir assemblé un puzzle dont les pièces étaient, nous semble-t-il, jusqu'ici manipulées isolément : le end-user programming en est une. Classiquement, la Méga-IHM ne laissait apparaître d'elle que la traditionnelle IHM et son ingénierie. L'Extra IHM est arrivée avec l'informatique ambiante, puis la Méta-IHM inspirée par l'IDM. Et là, le puzzle s'est assemblé.

Il nous faut, à présent, et avant d'aller plus loin, évaluer notre approche. Nous identifions deux points de vue : l'ingénierie d'IHM et l'enseignement de l'IHM. Pour l'ingénierie d'IHM, nous nous référons à [Myers 00] qui propose cinq critères d'évaluation d'outils :

- La couverture de l'IHM : nous couvrons aujourd'hui tous les aspects de l'IHM excepté le noyau fonctionnel. Intégrer le noyau fonctionnel à la réflexion, c'est traiter la plasticité des systèmes interactifs et non seulement de l'IHM. C'est une perspective au travail ;
- Le seuil (threshold) et le plafond (ceiling) : le seuil représente le coût d'entrée humain, c'est-à-dire l'effort à fournir pour être opérationnel avec l'outil. Le plafond est le capital apporté par l'outil. Brad Myers constate que les outils populaires sont soit à bas seuil et bas plafond, soit à fort seuil et haut plafond. Le défi est de produire des outils à bas seuil et fort potentiel. Si les approches à base de modèles font partie du cursus de tout ingénieur spécialisé en IHM, nous pensons pouvoir atteindre un faible seuil pour un fort potentiel : d'une part, remodelage et redistribution sont banalisés dans l'approche ; d'autre part, l'outil permet un prototypage rapide par l'unification entre la conception et l'exécution ;
- Le chemin de moindre résistance : par le maintien à l'exécution du cahier des charges et l'expression de la valeur dans les transformations, nous pensons l'approche proactive, guidant le manipulateur dans la voie la plus prometteuse ;
- La prédictibilité : les outils même partiellement automatiques ont reçu un accueil réservé des programmeurs. Ici, tout est laissé sous le contrôle de l'utilisateur, préservant donc l'approche de cet écueil ;
- Le changement de cible : l'écart de temps entre, d'une part, la compréhension d'un besoin (la tâche du programmeur) et la mise à disposition d'un outil peut, du fait d'évolutions technologiques rapides, déboucher sur un outil déjà obsolète. Nous espérons être en avance de phase ☺

D'un point de vue de l'enseignement de l'IHM, nous espérons, par cet outil, aider les étudiants à mieux comprendre les étapes du processus de conception d'IHM et la ventilation des critères d'ergonomie au fil de ce processus. Donc même si la plasticité s'averrait décevante à l'usage, nous pensons, par nos études, avoir poussé les acquis du passé dans leurs retranchements, rendu explicites des implicites et comblé des manques.

Au-delà des sujets naissants posés au chapitre III, les perspectives sont nombreuses. Nous les organisons autour de deux grands défis.

1. Intégration : Atelier de plasticité

Intégrer nos travaux en un atelier de plasticité est une perspective chère à l'équipe. Nous souhaiterions, en particulier, intégrer les travaux sur la multimodalité. Les modalités et combinaisons de modalités sont une extension fondamentale pour ouvrir la plasticité sur les techniques d'interaction innovantes. La mixité de l'approche sera ici précieuse pour l'intégration logicielle de composants existants. Un premier pas est aujourd'hui franchi avec les COMET.

2. Interaction : Usage et Ethique

Jusqu'ici, nos travaux ont été menés selon une perspective système, appliqués à des démonstrateurs "jouet". Une fois l'atelier de plasturgie suffisamment mature, notre ambition est de traiter de vraies applications pour palper la réelle valeur de la plasticité. Une question intéressante sera celle de l'incidence sur l'interaction. La tâche utilisateur et, de façon plus générale, la valeur attendue changera-t-elle sur changement de contexte d'usage ? L'hypothèse faite dans nos travaux est la stabilité de cette valeur. Mais la co-évolution entre l'utilisateur et le système interactif pourrait être une perspective à notre travail. Nous avons comme garde-fou l'Extra-IHM, permettant, si nécessaire, la redéfinition de la valeur.

Enfin, d'un point de vue personnel, je serais curieuse d'examiner la plasticité sous d'autres angles : en particulier, les mathématiques et la thermodynamique. En topologie, nous avons la notion de voisinage qui pourrait être pertinente pour raisonner sur les domaines de plasticité.

En thermodynamique, les optimums locaux, globaux, l'entropie, l'enthalpie sont autant de réminiscences qui pourraient aider à raisonner sur l'état d'un système.

Pour terminer, je me réfère à Victor Hugo qui disait : « Moi, j'ai toujours pitié du pauvre marbre obscur. De l'homme moins souvent, parce qu'il est plus dur ». En sept ans, j'ai eu l'immense chance de ne travailler, au contraire, qu'avec des Hommes plastiques et lumineux : je les en remercie.

Acronymes

BàO :	Boîte à Outils
COMET :	Context Mouldable WidgETs
COSM	Cobweb of Systems and Models
HHCS :	Home Heating Control System
IDM :	Ingénierie Dirigée par les Modèles
IHM :	Interface Homme-Machine
MARA :	Models At Runtime for Adaptation
MDE :	Model Driven Engineering
MUST :	Mixing & Unifying Systems with Transformations
OS :	Operating System
PDA :	Personal Digital Assistant
WIMP :	Window Icon Mouse Pointer
ZOOOMM :	ZOO Of MetaModels

Glossaire

A

Acquise :	La plasticité est dite acquise si elle n'est pas innée.
Adaptable :	Se dit d'un système personnalisable sur intervention explicite de l'utilisateur qui peut agir sur les paramètres fixés par le concepteur. Définition donnée par [Totterdell 90] et traduite par [Thevenin 01].
Adaptatif :	Se dit d'un système doué de discrimination. Il sait reconnaître la situation (parmi plusieurs déclencheurs fixés par le concepteur) et adopte la réaction (recommandation) fixée elle aussi par le concepteur pour cette situation. Mais le système est incapable d'évaluer l'effet de sa réaction. Définition donnée par [Totterdell 90] et traduite par [Thevenin 01].
Aristomorphe :	Du grec <i>aristos</i> : meilleur. Se dit d'une IHM polymorphe capable de choisir la forme la plus appropriée selon le contexte d'usage.
Auto-explicative :	Se dit d'une IHM autoconnaissante capable de communiquer sur cette connaissance.
Autoconnaissante :	Se dit d'une IHM qui connaît sa raison d'être. Chaque interacteur sait dans quel espace de dialogue il se trouve et à quelle tâche il correspond.
Automatique :	La plasticité est dite automatique si l'adaptation est réalisée sans le concours de l'utilisateur.
Automédiateur :	Se dit d'un système qui a les caractéristiques de l'autorégulation, mais en plus, est capable de résoudre le problème de l'adaptation par planification et peut évaluer a priori l'effet d'une réaction (alors qu'un système autorégulateur n'évalue l'effet qu'a posteriori). Définition donnée par [Totterdell 90] et traduite par [Thevenin 01].
Automodificateur :	Se dit d'un système doué de généralisation qui, par ses méta-connaissances, a la capacité d'apprendre de nouveaux déclencheurs et de nouvelles réactions. Définition donnée par [Totterdell 90] et traduite par [Thevenin 01].
Autonome :	La plasticité est dite autonome si l'IHM contient tous les mécanismes logiciels et ressources numériques nécessaires à l'adaptation.
Autorégulateur :	Se dit d'un système doué de facultés d'auto-évaluation. Non seulement il reconnaît la situation parmi les déclencheurs prévus par le concepteur, mais utilise, pour effectuer son choix de réaction, une fonction de feedback sur lui-même par essai-erreur. Autrement dit, il utilise une métrique qui évalue l'effet de la réaction retenue précédemment pour ce même déclencheur. Définition donnée par [Totterdell 90] et traduite par [Thevenin 01].

C

Câblé :	Se dit d'un système dont le comportement est défini à la conception. Aucune adaptation n'est possible. Définition donnée par [Totterdell 90] et traduite par [Thevenin 01].
Contexte d'usage :	Triplet <Utilisateur, Plate-forme, Environnement> où : <ul style="list-style-type: none">• L'utilisateur dénote l'utilisateur du système

interactif. Il peut être décrit par ses compétences métier, informatiques, etc. ainsi que par des données générales (âge, taille, etc.) ;

- La plate-forme cerne les requis matériels et logiciels nécessaires à l'interaction. Typiquement, les dispositifs d'entrée et de sortie sont à considérer ;
- L'environnement se réfère à l'espace physique hébergeant l'interaction. Il peut être décrit par ses conditions lumineuses, sonores, sociales, etc.

Contrôlée : La plasticité est dite contrôlée si elle n'est pas automatique.

D

Décrémentale : La plasticité est dite décrémenteale si la valeur V ou les contextes d'usage C diminuent à l'usage.

Dépendante : La plasticité est dite dépendante si elle n'est pas autonome.

Dévalorisante : Une plastification est dite dévalorisante si elle réduit la valeur de l'IHM.

Directionnelle : Forme dégradée de plasticité. La plasticité n'est plus définie sur un graphe complet de contextes d'usage, mais sur un chemin orienté de contextes d'usage.

Distribué : Caractère d'une IHM à être rendue sur au moins deux plates-formes, quels que soient les rôles tenus par ces plates-formes.

Domaine de plasticité : Le domaine de plasticité d'une IHM pour une valeur V est l'union des îlots de plasticité de cette IHM pour cette valeur V .

Formalisation par l'IDM et la théorie des graphes :

Le domaine de plasticité d'une IHM I pour une valeur V est l'ensemble des p graphes G_i non orientés complets tels que

$\forall i \in \{1..p\}, I$ est plastique pour V sur G_i

et $\forall i \in \{1..p\}, \exists j \in \{1..p\} / i \neq j$ et G_j soit sous-graphe de G_i

et $\exists G_k / G_k \notin \{G_1...G_p\}$ et I est plastique pour V sur G_k

E

Elastique : La plasticité est dite élastique si la valeur V ou les contextes d'usage C varient à l'usage.

Équivalente : Une plastification est dite équivalente si elle conserve la valeur de l'IHM.

Extra-IHM : IHM de la fonction d'adaptation. Elle place le contexte d'usage, le remodelage et la redistribution de l'IHM sous le contrôle de l'utilisateur.

Dans l'approche scientifique MUST, c'est l'IHM du graphe de modèles M1.

F

Forme : De façon réductive, la forme est assimilée à la présentation. En réalité, le polymorphisme pourrait aussi intervenir en abstraction.

I

Ilot de plasticité :	Pour une valeur V donnée, un îlot de plasticité est un ensemble de contextes d'usage C tel que l'IHM est maximale ment plastique en contexte d'usage pour V sur C .
Incrémentale :	La plasticité est dite incrémentale si la valeur V ou les contextes d'usage C augmentent à l'usage.
Innée :	La plasticité est dite innée si elle contient, par construction, tous les mécanismes logiciels et ressources numériques nécessaires à l'adaptation.
Interacteur :	Entité numérique offrant des fonctions de service, rendues observables et/ou manipulables à l'utilisateur par le biais d'entités physiques qui jouent le rôle de ressources d'interaction [Dâassi 07].

M

Maximale :	La plasticité est dite maximale si elle est maximale en valeur et en contexte d'usage. La plasticité est maximale en valeur s'il n'existe pas de propriété p non considérée dans V telle que l'IHM soit plastique pour $V \cup \{p\}$ sur C . La plasticité est maximale en contexte d'usage s'il n'existe pas de contexte d'usage c non considéré dans C tel que l'IHM soit plastique pour V sur $C \cup \{c\}$.
Méga-IHM :	IHM du mégamodèle liant les niveaux M1 et M2 du graphe de modèles.
Méta-IHM :	IHM du graphe des métamodèles de remodelage et redistribution d'IHM.
Minimale :	La plasticité est dite minimale (respectivement minimale en valeur et en contexte d'usage) si elle n'est pas maximale (respectivement maximale en valeur et en contexte d'usage).

P

Plasticité :	Propriété d'une IHM à être plastique.
Plastifiabilité :	Plasticité dépendante.
Plastifiable :	Se dit d'une IHM satisfaisant la propriété de plastifiabilité.
Formalisation par l'IDM et la théorie des graphes : Soit MMc un métamodèle de contexte d'usage. Soit G un graphe non orienté complet de sommets $C1 .. Cn$, n contextes d'usage conformes à MMc . Notons C cet ensemble de sommets. Soit V une valeur. Une IHM I est dite plastifiable sur G pour V si $\forall Ci \in C, I \text{ satisfait } V \text{ dans } Ci$ et $\forall Ci, Cj \in C / i \neq j, I \text{ est capable d'être adaptée au changement de contexte d'usage de } Ci \text{ à } Cj \text{ dans le respect de } V.$	
Plastification :	Acte de plastifier.
Plastifier :	Action d'adapter une IHM à son contexte d'usage dans le respect de la valeur attendue.
Plastique :	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 .

Formalisation par l'IDM et la théorie des graphes :

Soit MMc un métamodèle de contexte d'usage.

Soit G un graphe non orienté complet de sommets $C1 .. Cn$, n contextes d'usage conformes à MMc . Notons C cet ensemble de sommets.

Soit V une valeur.

Une IHM I est dite plastique sur G pour V si

$\forall Ci \in C, I$ satisfait V dans Ci

et $\forall Ci, Cj \in C / i \neq j, I$ est capable de s'adapter au changement de contexte d'usage de Ci à Cj dans le respect de V .

Polymorphe :

Se dit d'une IHM possédant plusieurs formes.

R

Rigide :

S'applique aux IHM et à la plasticité.

Une IHM est dite rigide pour une valeur V si elle n'est pas plastique pour cette valeur.

La plasticité est dite rigide si la valeur V et les contextes d'usage C sont stables à l'usage.

V

Valeur :

La valeur fait référence à un ensemble de propriétés centrées sur l'utilisateur et/ou le système. Le choix des propriétés et la définition de leurs domaines de valeurs est du ressort du concepteur.

Valorisante :

Une plastification est dite valorisante si elle augmente la valeur de l'IHM.

Références

- [Abowd 92] Abowd, G.D., Coutaz, J., Nigay, L. Structuring the Space of Interactive System Properties, Engineering for Human-Computer Interaction, Larson J. & Unger C. (eds), Elsevier Science Publishers B.V. (North-Holland), IFIP, 1992, pp 113-126
- [Antoniac 02] Antoniac, P., Pulli, P., Kuroda, T., Bendas, D., Hickey, S., Sasaki, H. Wireless User Perspectives in Europe : HandSmart Mediaphone Interface, Wireless Personal Communications, Vol. 22, pp. 161-174, 2002
- [Arch 92] Arch, A Metamodel for the Runtime Architecture of An Interactive System, The UIMS Developers Workshop, SIGCHI Bulletin, 24(1), ACM (1992)
- [Baldauf 04] Baldauf M., Dustdar S., A Survey on Context-aware systems, International Journal of Ad Hoc and Ubiquitous Computing, 2004
- [Balme 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, Volume 3295 / 2004, Ambient Intelligence: Second European Symposium, EUSAI 2004, Markopoulos P., Eggen B., Aarts E. et al. (Eds), Springer-Verlag Heidelberg (Publisher), ISBN: 3-540-23721-6, Eindhoven, The Netherlands, November 8-11, 2004, 291-302
- [Barralon 02] Barralon, N. Interfaces Homme-Machine de Transition, DEA Informatique Système et Communication, Université Joseph Fourier, Juin 2002
- [Bastien 93] Bastien, J.M.C., Scapin D. Ergonomic Criteria for the Evaluation of Human-Computer Interfaces, Rapport technique INRIA, N°156, Juin 1993
- [Beaudouin-Lafon 00] Beaudouin-Lafon, M. Instrumental Interaction: an Interaction Model for Designing Post-WIMP User Interfaces. In Proc. ACM Human Factors in Computing Systems, CHI 2000, La Haye (Pays-Bas), Avril 2000, CHI Letters 2(1):446-453, ACM Press
- [Bellavista 06] Bellavista, P., Corradi, A., Montanari, R., and Stefanelli, C. A mobile computing middleware for location- and context-aware internet data services. ACM Trans. Inter. Tech. 6, 4, Nov. 2006, 356-380
- [Berti 05] Berti, S., Paternò, F. Migratory multimodal interfaces in multidevice environments. In Proceedings of the International Conference on Multimodal Interfaces, ICMI'05, ACM Publ., 2005, pp 92-99
- [Blackwell 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, 2001, pp. 150-157
- [Browne 90] Browne, D., Totterdell, P., Norman, M. Adaptive User Interfaces, Academic Press, Computer And People Series, 1990
- [Brusilovsky 97] Brusilovsky, P., Schwarz, E. User as Student: Towards an Adaptive Interface for Advanced Web-Based Applications. In A. Jameson, C. Paris, and C. Tasso (Eds), UserModeling: Proceedings of the Sixth International Conference, UM97, pp. 177-188
- [Burnett 03] Burnett, M., Cook, C., Pendse, O., Rothermel, G., Summet, J., Wallace, C. End-user software engineering with assertions in the spreadsheet paradigm. In Proc. ICSE 2003, pp. 93-103
- [Calvary 98] Calvary, G. Proactivité et réactivité : de l'Assignation à la Complémentarité en Conception et Evaluation d'Interfaces Homme-Machine, Thèse de l'Université Joseph Fourier Grenoble I, Octobre 1998, 250 pages
- [Calvary 01] Calvary, G., Coutaz, J., Thevenin, D. A Unifying Reference Framework for the Development of Plastic User Interfaces, Proceedings of 8th IFIP International Conference on Engineering for Human-Computer Interaction EHCI'2001 (Toronto, 11-13 May 2001), R. Little and L. Nigay (eds.), Lecture Notes in Computer Science, Vol. 2254, Springer-Verlag, Berlin, 2001, pp. 173-192

- [Calvary 02] Calvary, G., Coutaz, J. Plasticité des Interfaces : une nécessité !, Information-Interaction-Intelligence, Actes des deuxièmes Assises nationales du GDR I3, Cépaduès Editions, J. Le Maître (Ed), Nancy, Décembre 2002, pp 247-261
- [Calvary 03] Calvary, G., Coutaz J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A unifying reference framework for multi-target user interfaces, *Interacting With Computers*, Vol. 15/3, pp 289-308, 2003
- [Calvary 06] Calvary, G., Coutaz, J., Dâassi, O., Ganneau, V., Balme, L., Demeure, A., Sottet, J-S. Métamorphose des IHM et Plasticité : Article de synthèse, 10ème onference ERGO-IA, L'humain comme facteur de performance des systèmes complexes, 11-13 Octobre 2006, E. Brangier, C. Kolski et J.R. Ruault (eds), ESTIA & ESTIA.INNOVATION, Biarritz, France, pp 79-86
- [Capra 03] Capra, L., Emmerich, W., Mascolo C., CARISMA: Context-Aware Reflective iddleware System for Mobile Applications, *IEEE Transactions on Software Engineering*, vol. 29, no. 10, pp. 929-945, Oct., 2003
- [Clerc 03] Clerc, X., Pellier, D., Fiorino, H. Planification multi-agent par cycles de conjecture/réfutation. In *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*, 2003, November, pages 67-79, Hammamet, Tunisie
- [Clerckx 04] Clerckx, T., Luyten, K., Coninx, K. The Mapping Problem Back and Forth: Customizing Dynamic Models while Preserving Consistency. In *Proc. of the 3rd Int. Workshop on Task Models and Diagrams for User Interface Design TAMODIA'2004* (Prague, November 15-16, 2004). ACM Press, New York, (2004), pp 33-42
- [Cockton 04] Cockton, G. From Quality in Use to Value in the World. In *ACM Proc. CHI 2004, Late Breaking Results*, 2004, pp 1287-1290
- [Cockton 05] Cockton, G. A development Framework for Value-Centred Design. In *ACM Proc. CHI 2005, Late Breaking Results*, 2005, pp 1292-1295
- [Constantine 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
- [Conway 97] Conway, M., Pausch, R. Alice: easy to learn interactive 3D graphics. *ACM SIGGRAPH Computer Graphics*, 31(3), 1997
- [Coutaz 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 conference*, S. A. Arnesen & D. Gilmore Eds., Chapman&Hall Publ., Lillehammer, Norway, June 1995, pp. 115-120
- [Coutaz 01] Coutaz, J. *Architectural Design for User Interfaces ; The Encyclopedia of Software Engineering*, J. Marciniak Ed., Wiley & Sons Publ., seconde édition, 2001
- [Coutaz 05] Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D. Context is Key, *Communications of the ACM March 2005/Vol. 48, No. 3*, pp 49-53
- [Coutaz 06] Coutaz, J. Meta-User Interfaces for Ambient Spaces, *Interational Workshop on Task Model and Diagram (TAMODIA'06)*, Bucarest, Romania, 2006
- [Crease 00] Crease, M, Brewster, S.A., Gray, P. Caring, sharing widgets : a toolkit of sensitive widgets, In, *14th Annual Conference of the British HCI Group*, 5-8 September 2000, Sunderland, England, British Computer Society conference series, pp. 257-270
- [Crowley 02] Crowley, J.L., Coutaz, J., Rey, G., Reignier, P. Perceptual Components for Context-Aware Computing, *UbiComp 2002 :Ubiquitous Computing*, 4th International Conference, Göteborg, Sweden, Sept./Oct. 2002, G. Borriello, L.E. Holmquist Eds., LNCS, Springer Publ., pp. 117-134
- [Cypher 93] Cypher, A. *Whatch What I Do*. The MIT Press, 1993
- [Dâassi 07] Dâassi, O. Les comets : une nouvelle génération d'interacteurs pour la plasticité des Interfaces Homme-Machine. Thèse de l'Université Joseph Fourier, Grenoble

- I, Janvier 2007
- [David 06] David, P.-C., Ledoux, T. An Aspect-Oriented Approach for Developing Self-Adaptive Fractal Components. In: 5th Int. Symposium on Software Composition. Vienna, Austria, 2006
- [Demeure 07] Demeure, A. Modèles et Outils pour la Conception et l'Exécution d'Interfaces Homme-Machine Plastiques, Thèse de l'Université Joseph Fourier, 11 Octobre 2007
- [Dey 00] Dey, A. Providing Architectural Support for Building Context-Aware Applications, PhD thesis, College of Computing, Georgia Institute of Technology, December 2000
- [Dey 04] Dey, A., Hamid, R., Beckmann, C., Li, I., Hsu, D. a CAPpella: Programming by demonstration of context-aware applications. In Proc. ACM Conference on Human Factors in Computing Systems (CHI), 2004
- [Dieterich 94] Dieterich, H., Malinowski, U., Kühme, T., Schneider-Hufschmidt, M. State of the Art in Adaptive User Interfaces, in Adaptive User Interfaces : Principles and Practice, Schneider-Hufschmidt & Al.(ed.), 1994, pp. 13-48
- [Dix 93] Dix, A., Finlay, J., Abowd, G., Beale, R. Human-Computer Interaction, Prentice-Hall, New-Jersey, 1993
- [Dubois 01] Dubois, E. Chirurgie Augmentée, un cas de Réalité Augmentée. Conception et réalisation centrées sur l'utilisateur. Thèse de doctorat Informatique de l'Université Joseph Fourier, Juillet 2001, 275 pages
- [Favre 04a] Favre, J.M. Foundations of Model (Driven) (Reverse) Engineering: Models - Episode I, Stories of the Fidus Papyrus and of the Solarus, in Dagstuhl Seminar 04101 on Language Engineering for Model-Driven Software Development, Dagstuhl, Germany, February 29-March 5, 2004, DROPS proceedings, <http://drops.dagstuhl.de/portals/04101/>
- [Favre 04b] Favre, J.M. Foundations of the Meta-pyramids: Languages and Metamodels – Episode II, Story of Thotus the Baboon, in Dagstuhl Seminar 04101 on Language Engineering for Model-Driven Software Development, Dagstuhl, Germany, February 29-March 5, 2004, DROPS proceedings, <http://drops.dagstuhl.de/portals/04101/>
- [Favre 04c] Favre, J.M. Towards a Basic Theory to Model Model Driven Engineering, Workshop on Software Model Engineering, WISME @ UML2004, Lisboa, Portugal, October 11, 2004, <http://www-adele.imag.fr/~jmfavre>
- [Favre 04d] Favre, J.M., Nguyen, T. Towards a Megamodel to Model Software Evolution Through Software Transformation, Workshop on Software Evolution through Transformation, SETRA 2004, Rome, Italy, October 2, 2004, Electronic Notes in Theoretical Computer Science, Volume 127, Issue 3, ENTCS ELSVIER
- [Fekete 05] Fekete, J.D. Nouvelle génération d'Interfaces Homme-Machine pour mieux agir et mieux comprendre, Habilitation à diriger des recherches, Spécialité Informatique, INRIA Futurs et Laboratoire de Recherche en Informatique (LRI), Orsay, 2 mai 2005
- [Findlater 04] Findlater, L., McGrenere, J. A comparison of static, adaptive, and adaptable menus, In Proceedings of CHI'04, ACM Press, 2004, pp 89–96
- [Fischer 04] Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N. Meta-design: a manifesto for end-user development, Communications of the ACM, Volume 47 , Issue 9 (September 2004), End-user development: tools that empower users to create their own software solutions, Special Issue: End-user development, ACM Press, 2004, pp 33-37
- [Florins 06] Florins, M. Graceful Degradation, A Method for Designing Multiplatform Graphical User Interfaces, PhD Université Catholique de Louvain, Canada, Juillet 2006
- [Ganneau 07] Ganneau, V., Calvary, G. Demumieux, R. Métamodèle de Règles d'Adaptation pour la Plasticité des Interfaces Homme-Machine, IHM'07, A paraître

- [Girard 92] Girard, P. Environnement de programmation pour non programmeurs et paramétrage en conception assistée par ordinateur : Le système Like. Thèse HDR, Nov. 1992. Univ. Poitiers
- [Griffiths 01] Griffiths, T., Barclay, P.J., Paton, N.W., McKirdy, J., Kennedy, J.B., Gray, P.D., Cooper, R., Goble, C.A., da Silva, P. Teallach: a Model-Based User Interface Development Environment for Object Databases. *Interacting with Computers* 14, 1, 2001, pp 31-68
- [Grolaux 04]] Grolaux, D., Van Roy, P., Vanderdonckt, J. Migratable User Interfaces : Beyond Migratory Interfaces, in *Proc. Of Mobiquitous 2004 The First Annual International Conference on Mobile and Ubiquitous Systems Networking and Services*, 2004, Boston, August 22-26
- [Grolaux 05] Grolaux, D., Vanderdonckt, J., Van Roy, P. Attach Me, Detach Me, Assemble Me Like You Work, In *proceedings of INTERACT 2005*, M.F. Costabile and F. Paternò (Eds.), 2005, pp 198 – 212
- [IFIP 96] Design Principles for Interactive Software, livre produit par l'IFIP WG 2.7 (13.4), C. Gram & G. Cockton Eds., Chapman&Hall Publ., 1996
- [Jabarin 03] Jabarin, B., Graham, N., Architectures for Widget-Based Plasticity, in *Proceedings of Design, Specification and Verification of Interactive Systems (DSV-IS 2003)*, Springer LNCS, 2003, pp 124-138
- [Johnson 93] Johnson, J., Nardi, B., Zarnier, C., Miller, J. ACE : building interactive graphical applications, *Communication ACM*, volume 36, number 4, ACM Press, New York, NY, USA, 1993, pp 40-55
- [Karsenty 05] Karsenty, L., Botherel, V. Analyse empirique de l'inter-utilisabilité d'un service multisupport Web et onferenc. *Actes des Deuxièmes Journées Francophones sur l'Ubiquité et la Mobilité (UbiMob'05)*, Grenoble, 31 Mai-3 Juin 2005, ACM Press
- [Kawai 96] Kawai, S., Aida, H., Saito, T. Designing interface toolkit with dynamic selectable modality, In *Proceedings of the Second Annual ACM Conference on Assistive Technologies (Vancouver, British Columbia, Canada, April 11-12, 1996, Assets '96*, ACM Press, New York, NY, pp 72-79
- [Kephart 03] Kephart, J.O., Chess, D. The Vision of Autonomic Computing, *Computer*, vol. 36, no. 1, 2003, pp. 41–50
- [Kiczales 97] Kiczales, G. et al. Aspect-oriented programming, *Proc. Euro. Conf. OOP 1997*, LNCS 1241, Springer Verlag, 1997
- [Lachenal 04] Lachenal, C. Modèle et infrastructure logicielle pour l'interaction multi-surface multi-instrument. Thèse de doctorat Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (CLIPS), Université Joseph Fourier, Décembre 2004
- [Lecolinet 03] Lecolinet, E. A molecular architecture for creating advanced GUIs, In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology, UIST '03*, Vancouver, Canada, November 02-05, 2003, ACM Press, New York, NY, pp 135-144
- [Lepreux 06] Lepreux, S., Vanderdonckt, J. Towards Supporting User Interface Design by Composition Rules, *Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'06 (Bucharest, 6-8 June 2006)*, Springer-Verlag, Berlin, 2007, pp. 231–244
- [Letondal 05] Letondal, C. Participatory Programming: Developing programmable bioinformatics tools for end-users. In H. Lieberman, F. Paterno, & V. Wulf (Eds.), *End-User Development*. Springer/Kluwer Academic Publishers
- [Limbourg 04a] Limbourg, Q. Multi-path Development of User Interfaces, Ph.D. thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 4 November 2004
- [Limbourg 04b] Limbourg, Q., Vanderdonckt, J. Addressing the Mapping Problem in User Interface Design with UsiXML. In *Proc. of the 3rd Int. Workshop on Task Models and Diagrams for User Interface Design TAMODIA'2004 (Prague, November 15-*

- 16, 2004). ACM Press, New York, 2004, pp 155-163
- [Little 07] Little, G., Lau, T., Lin, J., Kandogan, E., Haber, E., Cypher, A. Koala: Capture, Share, Automate, Personalize Business Processes n the Web. In Proc. CHI 2007, à paraître
- [Liu 03] Liu, J., Wong, C.H., Hui, K.K. An Adaptive User Interface Based On Personalized Learning. In IEEE Intelligent Systems 18 (2) 2003, pp. 52-57
- [Lopes 95] Lopes, C.V., Hursch, W.L. Separation of Concerns, College of Computer Science, Northeastern University, Boston, February 1995
- [Lopez-Jaquero 04] Lopez-Jaquero, V., Montero, F., Molina, J.P., Gonzalez, P. A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties, EHCI04
- [MacLean 96] MacLean, A., Young, R.M., Bellotti, V.M.E. and Moran, T.P. Questions, Options, and Criteria: Elements of Design Space Analysis. In Moran, T.P., Carroll, J.M., eds Design Rationale: Concepts, Techniques, and Use. LEA. 1996
- [Maes 87] Maes, P. Concepts and experiments in computational reflection, OOPSLA'87, siacmsn no22-12, 1987, p. 147-155
- [Mahéo 04] Mahéo, Y., Guidec, F., Courtrai, L. Middleware Support for the Deployment of Resource-Aware Parallel Java Components on Heterogeneous Distributed Platforms. In 30th Euromicro Conference – Component-Based Software Engineering Track, IEEE Computer Society, Rennes, France, pp 144-151
- [Martin 95] Martin, J., Crowley, J.L. Comparison of Correlation Techniques. In Intelligent Autonomous Systems, pages 86-93, 1995
- [Montero 04] Montero, F., Vanderdonckt, J., Lozano, M. Quality Models for Automated Evaluation of Web Sites Usability and Accessibility, In proc. Of the International Conference on Web Engineering, ICWE'2004, July 28-30, Munich, 2004
- [Mori 02] Mori, G., Paternò, F., Santoro C. CTTE : Support for Developing and Analyzing Task Models for Interactive System Design, IEEE Transactions on Software Engineering August 2002, pp.797-813.
- [Myers 00] Myers, B., Hudson, S.E., Pausch, R. Past, Present, and future of user interface software tools, ACM Transactions on Computer-Human Interaction (TOCHI), Volume 7, Issue 1 (March 2000), Special issue on human-computer interaction in the new millennium, Part 1, pp 3 – 28, 2000
- [Myers 06] Myers, B., Weitzman, D.A., Ko, A.J., Chau, D.H. Answering why and why not questions in user interfaces, Proceedings of the SIGCHI conference on Human Factors in computing systems CHI'06, Montréal, Québec, Canada, 2006, pp 397-406
- [Newman 02] Newman, M.W., Sedivy, J.Z., Neuwirth, C.M., Edwards, W.K., Hong, J.I., Izadi, S., Marcelo, K., Smith, T.F, Designing for Serendipity: Supporting End-User Configuration of Ubiquitous Computing Environments. In Proc. Of Designing Interactive Systems (DIS), London 2002, 147-156
- [Nielsen 93] Nielsen, J. Usability Engineering, Academic Press Professional, 1993, 362 pages
- [Nielsen 94] Nielsen, J. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY, 1994. Also available at http://www.useit.com/papers/heuristic/heuristic_list.html
- [Nogier 05] Nogier, J.F. Ergonomie du logiciel et design Web : le manuel des interfaces utilisateur, 3^{ème} édition, Dunod, 272 pages, 2005
- [Normand 92] Normand, V. Le modèle SIROCO : de la spécification conceptuelle des interfaces utilisateur à leur réalisation. Thèse de l'Université Joseph Fourier-Grenoble I, Spécialité Informatique, Avril 1992, 258 pages
- [Oreizy 99] Oreizy, P., Gorlik, M., Tay lor, R., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D., Wolf, A. An Architecture-Based Approach to Self-Adaptive Software. In IEEE Intelligent Systems, May-June, 1999, pp 54-62
- [Paganelli 02] Paganelli, L., Paternò, F. Automatic Reconstruction of the Underlying Interaction Design of Web Applications, Proceedings of SEKE 2002, Ischia, Italy

- [Pane 02] Pane, J.F. A programming system for children that is designed for usability, PhD Thesis, Carnegie Mellon University
- [Paterno' 97] Paterno', F., Mancini, C., Meniconi, S. ConcurTaskTrees : A Diagrammatic Notation for Specifying Task Models. In Proceedings Interact'97, July'97, Sydney, Chapman&Hall, 1997, 362-369
- [Preece 94] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T. Human-Computer Interaction, Wokingham, UK, Addison Wesley Publ., 1994
- [Rashid 04] Rashid, A., Kortuem G. Adaptation as an aspect in pervasive computing, Symposium on Principles of Distributed Computing, 2004
- [Rekimoto 99] Rekimoto, J., Saitoh, M. Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments, Proceedings of CHI'99, ACM Press, 1999, pp. 378-385
- [Repenning 06] Repenning, A. Ioannidou, A. Agent-Based End-user Development. Communications of the ACM, 47(9), September 2004, pp. 43-46
- [Rey 05] Rey, G. Contexte en Interaction Homme-Machine : le contexteur, Thèse de l'Université Joseph Fourier, Grenoble I, 1^{er} août 2005. 186 pages
- [Rodden 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, August 1st-4th, Cambridge, Massachusetts: ACM Press
- [Salber 95] Salber, D. De l'interaction individuelle aux systèmes multi-utilisateurs. L'exemple de la Communication Homme-Homme-Médiatisée. Thèse de doctorat Informatique préparée au Laboratoire de Génie Informatique (IMAG), Université Joseph Fourier, 8 septembre 1995, 303 pages
- [Schmidt 99] Schmidt, A. Implicit human-computer interaction through context. 2th Workshop on Human Computer Interaction with Mobile Devices. Edinburgh, Scotland, 31 August 1999
- [Schmidt 00] Schmidt, D. Pattern-Oriented Software Architecture, Vol. 2 : Patterns for Concurrent and Networked Objects. John Wiley & Sons, 2000
- [Schneiderman 97] Schneiderman, B. Designing User Interface Strategies for effective Human-Computer Interaction (3rd ed), Addison-Wesley Publ., 1997, 600pp
- [Scholtz 99] Scholtz J., Muller M., Novick D., Olsen D.R., Schneiderman B., Wharton C. A Research Agenda for Highly Effective Human-Computer Interaction : Useful, Usable, and Universal, SIGCHI bulletin, ACM/SIGCHI, October 1999, Volume 31, Number 4, pp 13-16
- [Schuler 93] Schuler, D., Namioka, A. Participatory design : Principles and practices. Hillsdale, NJ : Erlbaum, 1993
- [Schulte 96] Schulte, R.W., Natis, Y.V. Service Oriented Architecture, (SPA-00-7425), Gartner, 1996
- [Seffah 04] Seffah, A., Donyaee, M., Kline, R. B. Usability and quality in use measurement and metrics : An integrative model. Software Quality Journal, 2004
- [Shackel 91] Shackel, B. Usability-Context, Framework, Design and Evaluation. In Human Factors for Informatics Usability, Cambridge University Press, 1991, pp 21-38
- [Smith 77] Smith, D. C., Pygmalion : A Computer Program to Model and Stimulate Creative Thought. Basel, Stuttgart, Birkhauser Verlag, 1977
- [Sohn 06] Sohn, T.Y., Dey, A.K., iCAP: An Informal Tool for Interactive Prototyping of Context-Aware Applications. In Proc. Of the International Conference on Pervasive Computing 2006. Dublin, Ireland, May 2006, 974-975
- [Sottet 05a] Sottet, J.S. M2R Informatique de l'Université Joseph Fourier, Grenoble I, Juin 2005
- [Sottet 05b] Sottet, J.S., Calvary, G., Favre, J.M. Ingénierie de l'Interaction Homme-Machine Dirigée par les Modèles, Premières Journées sur l'Ingénierie Dirigée par les Modèles, IDM'05, Sébastien Gérard, Jean-Marie Favre, Pierre-Alain Muller, Xavier Blanc, Editors, Paris, 30 juin-1er juillet 2005, ISBN 2-7261-1284-6, pp 67-

- [Sottet 06] Sottet, J.S., Calvary, G., Favre, J.M., Coutaz, J. IHM&IDM : Un tandem prometteur, Article court à ERGO'IA 2006, 10ème onference ERGO-IA, L'humain comme facteur de performance des systèmes complexes, 11-13 Octobre 2006, Biarritz, France, A paraître
- [Sottet 07a] Sottet, J.S., Calvary, G., Favre, J.M., Coutaz, J. Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces : Mega-UI, CHISE II book, Seffah, A. & Vanderdonckt, J. (Eds), Springer, 2007, A paraître.
- [Sottet 07b] Sottet, J.S., Calvary, G., Coutaz, J., Favre, J.M. A Model-Driven Engineering Approach for the Usability of Plastic User Interfaces, In the proceedings of Engineering Interactive Systems 2007 joining Three Working Conferences : IFIP WG2.7/13.4 10th Conference on Engineering Human Computer Interaction, IFIP WG 13.2 1st Conference on Human Centred Software Engineering, DSVIS – 14th Conference on Design Specification and Verification of Interactive Systems, University of Salamanca, Spain, March 22-24, 2007
- [Sottet 07c] Sottet, J.S., Calvary, G., Coutaz, J., Favre, JM, Vanderdonckt, J., Stanciulescu, A., Lepreux, S. Language engineering : One does not fit all, Journal on Multimodal User Interfaces, Vol. 2, A paraître
- [Sottet 07d] Sottet, J.S., Ganneau, V., Calvary, G., Coutaz, Demeure, A., Favre, J.M., Demumieux, R. Model Driven Adaptation for Plastic User Interfaces, Interact'07, Rio de Janeiro, September 10-14, A paraître
- [Stuerzlinger 06] Stuerzlinger, W., Chapuis, O., Phillips, D., Roussel, N. User Interface Façades : Towards Fully Adaptable User Interfaces, In Proceedings of UIST'06, the 19th ACM Symposium on User Interface Software and Technology, ACM Press, 2006, pp 309-318
- [Szyperski 98] Szyperski, C. Component Software, Beyond Object-Oriented Programming, ACM, 1998
- [Suchman 87] Suchman, L. Plans and Situated Actions, Cambridge University Press, 1987
- [Szekely 96] Szekely, P. Retrospective and Challenges for Model-Based Interface Development, Proceedings of CADUI'96 (Computer-Aided Design of User Interfaces), J. Vanderdonckt (eds), Presses Universitaires de Namur, 1996, pp xxi-xliv
- [Thevenin 99] Thevenin, D., Coutaz, J. Plasticity of User Interfaces : Framework and Research Agenda. In Proc. Interact99, Edinburgh, A. Sasse & C. Johnson Eds, IFIP IOS Press Publ. , 1999, pp.110-117
- [Thevenin 01] Adaptation en Interaction Homme-Machine : le cas de la Plasticité. Thèse de l'Université Joseph Fourier, Grenoble I, 2001, 234 pages
- [Tigli 06] Tigli, J.Y., Cheung-Foo-Wo, D., Lavirotte, S., Riveill, M. Adaptation au contexte par tissage d'aspects d'assemblage de composants déclenchés par des conditions contextuelles, RTSI Série ISI – Adaptation et Gestion du Contexte, volume 11, numéro 5, pp 89-114, 2006. 2-7462-1672-8
- [Totterdell 90] Totterdell, P., Rautenbach, P. Adaptation as a Problem Design, in Adaptive User Interfaces. Paru dans [Browne et al. 1990a], pp. 59-84.
- [Trevisan 03] Trevisan, D., Vanderdonckt, J., Macq, B. Continuity as a usability property, HCI 2003 – 10th Intl Conference on Human-Computer Interaction, Heraklion, Greece, June 22-27, 2003, Vol. I, pp. 1268-1272, 2003
- [url CAMELEON] <http://giove.cnuce.cnr.it/cameleon.html>
- [url UsiXML] <http://www.usixml.org>
- [url ZOOMM] <http://zoomm.org/>
- [Van Welie 99] Van Welie, M., van der Veer, G.C., Eliëns, A. Usability Properties in Dialog Models: In 6th International Eurographics Workshop on Design Specification and Verification of Interactive Systems DSV-IS99, Braga, Portugal, 2-4 June (1999) 238-253
- [Vanderdonckt 05] Vanderdonckt, J., Grolaux, D., Van Roy, P., Limbourg, Q., Macq, B., Michel,

B., A Design Space for Con-text-Sensitive User Interfaces, Proc. of ISCA
14th Int. Conf. on Intelligent and Adaptive Systems and Soft-ware
Engineering IASSE,2005 (Toronto, July 20-22, 2005), 2005
[Winograd 01] Winograd, T. Architecture for Context, Human Computer Interaction, Vol. 16,
2001, pp 401-419

Annexe A : Curriculum vitae

Annexe B : Sélection d'articles de recherche

Auteurs : **Calvary**, G., Coutaz J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.

Titre : A Unifying Reference Framework for Multi-Target User Interfaces

Journal : Interacting With Computers, Vol. 15/3, pp 289-308, 2003

Auteurs : Demeure, A., **Calvary**, G., Coutaz, J., Vanderdonckt, J.

Titre : The Comets Inspector: Towards Run Time Plasticity Control based on a Semantic Network

Conférence : Fifth International Workshop on Task Models and Diagrams for UI design (TAMODIA'06), Hasselt, Belgium, October 23-24, 2006, pp 324-338

Auteurs : Sottet, J.S., Ganneau, V., **Calvary**, G., Coutaz, Demeure, A., Favre, J.M., Demumieux, R.

Titre : Model Driven Adaptation for Plastic User Interfaces

Conférence : Interact'07, Rio de Janeiro, September 10-14 2007