

Comet : Une nouvelle génération de « Widget » pour la Plasticité des Interfaces

Olfa Dâassi

Gaëlle Calvary

Joëlle Coutaz

Alexandre Demeure

CLIPS-IMAG
BP 53

38041 Grenoble Cedex 9

{Olfa.Daassi@imag.fr, Gaëlle.Calvary@imag.fr, Joëlle.Coutaz@imag.fr, Alexandre.Demeure@imag.fr}

RESUME

Avec l'informatique diffuse, les approches basées modèles sont aujourd'hui revisitées pour la production d'Interfaces Homme-Machine (IHM) plastiques, c'est-à-dire d'IHM capables de s'adapter à leur contexte d'usage dans le respect de leur utilisabilité. Par contexte d'usage, on entend un triplet <utilisateur, plate-forme, environnement>. Un cadre de référence en plasticité des interfaces organise l'adaptation des IHM autour de quatre niveaux d'abstraction, allant des concepts et tâches du domaine jusqu'à des ajustements physiques de l'interaction. Cet article décline ce cadre de référence à la granularité des interacteurs pour la production de « widgets » doués de faculté d'adaptation. Les *comets* (Context sensitive Multi-target widgETS) sont une nouvelle génération de widgets, capables de s'adapter à leur contexte d'usage. L'adaptation s'y opère à quatre niveaux d'abstraction : concepts et tâches, interfaces abstraites, concrètes et finales. Cet article couvre la conception et l'exécution de *comets*. Il en examine notamment l'architecture logicielle et les mécanismes d'adaptation. Ces mécanismes sont régis par un ensemble de stratégies et politiques.

MOTS CLES : Plasticité des interfaces, adaptation, contexte d'usage, multi-ciblage, widget, interacteur.

ABSTRACT

Adaptation of User Interfaces (UI) to context of use is becoming a major issue in HCI. By context of use, we mean the <user, platform, environment> triplet. Plasticity refers to the ability of UIs to withstand variations of context of use while preserving usability. Based on a reference framework that provides designers with a powerful tool for reasoning about plasticity, this paper addresses plasticity at the level of interactors. Comets (Context sensitive Multi-target widgETS) are a new

kind of widget that support adaptation at any level of abstraction: concepts and tasks, abstract, concrete and final UIs. This paper proposes a model as well as run-time software mechanisms that support the design of comets as well as run-time adaptation. Run-time is ruled according to a set of strategies and politics.

KEYWORDS : Plasticity of user interfaces, adaptation, context of use, multi-targeting, widget, interactor.

INTRODUCTION

Cet article fait suite à une communication informelle présentée à IHM'02 [5]. Il traite de la plasticité des Interfaces Homme-Machine (IHM), c'est-à-dire de leur capacité à s'adapter à leur contexte d'usage dans le respect de leur utilisabilité [13]. Par contexte d'usage, on entend un triplet <utilisateur, plate-forme, environnement> où :

- L'utilisateur dénote l'utilisateur du système interactif. Il peut être décrit par ses compétences dans le domaine applicatif, sa familiarité avec les systèmes interactifs, etc.
- La plate-forme cerne les requis matériels et logiciels nécessaires à l'interaction. Typiquement, la surface d'affichage et la présence de souris ou boule roulante 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.

Avec la diversité des plates-formes, le succès des ordinateurs de poche et leur usage en conditions mobiles, l'adaptation des interfaces à leur contexte d'usage est devenue un sujet d'actualité. Jusqu'aux travaux de Crease [4], l'adaptation était étudiée au grain des systèmes interactifs. En particulier, ArtStudio [14], Teresa [11] et leurs précurseurs Mastermind, Tadeus, FUSE ou TRIDENT (cf [15]) produisaient des IHM par assemblage d'objets d'interaction *rigides*, c'est-à-dire non doués de faculté d'adaptation.

Cet article traite d'objets d'interaction *multi-cibles* vs *plastiques*, c'est-à-dire d'objets d'interaction doués de faculté d'adaptation, préservant (plasticité) ou non (multi-ciblage) leur utilisabilité au cours de l'adaptation.

Copyright© 2003 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page or initial screen of the document. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. IHM 2003 November 25-28, 2003, Caen, France.
Copyright 2003 ACM 1-58113-803-2/03/0011...\$5.00

Baptisée *comet* (COntext sensitive Multi-target widgET), cette nouvelle génération de “widgets” est illustrée sur un cas d’étude (section 1) et motivée par une revue critique de l’état de l’art (section 2). Elle est examinée sous l’angle de la conception (section 3) et de l’exécution (section 4). Sa mise en oeuvre est évoquée en section 5. Elle donne lieu à un ensemble de perspectives énoncées en section 6.

CAS D’ETUDE

Considérons un usager souhaitant connaître l’état de ses comptes bancaires. Plusieurs possibilités s’offrent à lui, en particulier, une consultation sur distributeur bancaire (Figure 1a) ou connexion Internet (Figure 1b).

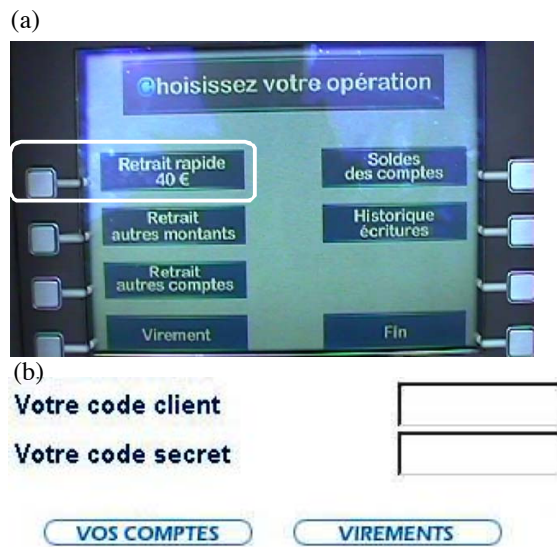


Figure 1 : Opérations bancaires sur distributeur (cas a) vs Internet (cas b). On constate que les IHM varient en termes de services, d’agencement, de formulation et d’incarnation. On notera en a) sur le distributeur l’ajout de traits blancs pour matérialiser le couplage entre les mondes physique et numérique.

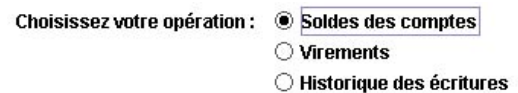
Les IHM présentées en Figure 1 diffèrent par quatre aspects :

- Les tâches ne sont pas les mêmes. En particulier, le retrait d’argent est impossible sur Internet.
- L’agencement des tâches varie, intervertissant entre les deux plates-formes les tâches de virement et de consultation des comptes.
- La formulation des tâches diffère, variant de « Soldes des comptes » à « VOS COMPTES ».
- L’incarnation des tâches est différente, en particulier dans son couplage entre les mondes physique et numérique. Alors qu’en b), un seul bouton physique suffit (celui de la souris) à l’activation de tout service (« VOS COMPTES » ou « VIREMENTS »), en a) deux colonnes de boutons physiques sont requises. Un bouton correspond à une tâche, mais n’est

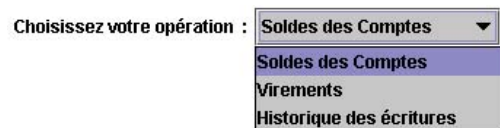
pas gravé du nom de cette tâche, les pages d’écran se succédant au fil de l’interaction. Aussi, des traits blancs sont-ils physiquement rajoutés pour forcer la perception du couplage entre les mondes physique (boutons) et numérique (libellés). En b), ce couplage est rendu observable par le pointeur souris. Lorsque ce pointeur arrive sur le libellé d’une tâche, il en permet l’activation. Cette balance entre les mondes physique et numérique se retrouve pour l’identification de l’usager. Alors qu’elle est véhiculée par la carte de crédit en a), elle est assurée par un code virtuel en b).

La figure 2 présente des variantes de la Figure 1b. Elles requièrent les mêmes dispositifs (une surface d’affichage et un dispositif de pointage), mais s’en distinguent par la représentation : les boutons virtuels de la Figure 1b sont ici remplacés par des libellés, boutons radio et menus déroulants. Au-delà du rendu, ces alternatives diffèrent par leurs empreintes graphiques et propriétés. Alors que les services sont directement observables en Figures 1b et 2a, ils sont parcourables en 2b et 2c.

(a) Libellé + boutons radio



(b) Libellé + menu déroulant



(c) Menu déroulant

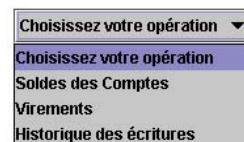


Figure 2 : Trois incarnations possibles de la tâche « Choisir une opération bancaire ».

Dans un objectif d’adaptation au contexte d’usage, l’idée est ici de percevoir ces IHM (Figures 1a, 1b, 2a, 2b, 2c) comme des manifestations différentes d’une même abstraction : la notion d’activation de service. Il s’agit, par le biais, des *comets* d’abstraire les widgets classiques pour rassembler autour d’une même abstraction un éventail de présentations. Cette vision des widgets annonce un nouveau type de boîtes à outils, recentrées sur la notion de tâche utilisateur. Elle allie deux thèmes de recherche, les interacteurs et la plasticité, dont nous présentons l’état de l’art dans la section suivante.

ETAT DE L'ART

Cette section examine l'état de l'art en vue de la conception et de l'exécution de *comets*. Elle considère les travaux sur les interacteurs et la plasticité puis énonce l'approche adoptée.

Interacteurs

La notion d' « interacteur » a été introduite par Faconti et Paterno en 1990 comme étant l'abstraction d'une entité capable à la fois d'entrée et de sortie dans un système interactif graphique [8]. Dès 1993, Duke et Harrisson notent que cette notion est intéressante pour raisonner sur les systèmes interactifs en général et que la définition doit donc être revue pour être portée au-delà du graphique [7]. Depuis, plusieurs contributions à dominante ingénierie (par opposition à l'approche analytique de Duke) ont été apportées, en particulier :

- Vanderdonck [15] qui, par sa distinction entre OIC (Objets d'Interaction Concrets) et OIA (Objets d'Interaction Abstraits) fait émerger deux niveaux d'abstraction dans la notion d'interacteur. Il identifie six classes d'OIAs : les OIAs statiques, les OIAs d'action, de défilement, de contrôle, de dialogue et de feed-back.
- Markopoulos [10] qui, dans la lignée de Duke [7], adopte une approche flux de données permettant la composition d'interacteurs. Il propose le modèle d'architecture ADC (Abstract-Display-Control) faisant émerger deux unités : un contrôleur, d'une part, un bloc Abstraction-Présentation, d'autre part.
- Thevenin [14] qui recentre les interacteurs sur les concepts et tâches du domaine. Un interacteur est décrit par « un nom ; la liste des données qu'il est capable de représenter ; la liste des tâches qu'il propose ; ainsi que son empreinte graphique ».
- Crease [4] qui dote les interacteurs de faculté d'adaptation en sortie. L'interacteur devient multimodal, évoluant dans les mondes graphique et sonore.

De cette revue de l'état de l'art, nous retenons que :

- si Crease innove en dotant les interacteurs de faculté d'adaptation, cette capacité se limite à la sortie et ne couvre pas la réalité mixte ;
- la qualité de service des présentations n'est pas modélisée. Seul Thevenin amorce la réflexion en chiffrant les requis de l'interacteur en termes de surface d'affichage ;
- les travaux se complètent en termes d'architecture logicielle montrant l'intérêt d'un contrôleur (ADC et Crease) et d'un raisonnement à différents niveaux d'abstraction (Vanderdonck et Thevenin).

Examinons maintenant l'état de l'art en matière de plasticité.

Plasticité

Dans l'espace problème de la plasticité, Calvary [1] perçoit deux leviers d'adaptation :

- un remodelage de l'interface. Par exemple, pour la tâche « Choisir opération bancaire » (Figure 2), commuter de la présentation b) à c) pour s'accommoder d'une largeur de surface d'affichage de plus en plus restreinte ;
- une redistribution de l'interface sur l'ensemble des plates-formes disponibles dans l'environnement. C'est la métaphore du peintre de Rekimoto [12] où la palette et la zone de dessin sont respectivement hébergées par un PDA (Personal Digital Assistant) et un PC. Pour notre exemple de distributeur bancaire, on pourrait envisager que sur le PDA soient aspirés tous les services opérables à distance, les autres restant sur le distributeur physique. On pourrait aussi envisager que l'accès à ces services reste sur PDA, mais que le rendu se fasse sur PC pour bénéficier d'une plus large surface d'affichage. Cette redistribution partielle (ou migration partielle) serait typiquement utile pour la tâche « Historique des écritures », consommatrice en surface d'affichage.

Les deux leviers, remodelage et redistribution, sont couverts, du point de vue de l'ingénierie, par un même cadre de référence en plasticité [2]. Ce cadre de référence unifie la conception et l'exécution d'IHM plastiques (Figure 3).

Conception. La conception s'articule autour d'un ensemble de modèles initiaux, c'est-à-dire spécifiés par le concepteur : les modèles du domaine (concepts et tâches utilisateur), du contexte d'usage (utilisateur, plate-forme, environnement) et de l'adaptation (évolution, transition). Le modèle d'évolution [6] spécifie la réaction à mettre en oeuvre en cas de changement de contexte d'usage. Le modèle de transition spécifie d'éventuelles mesures complémentaires permettant d'assurer une meilleure continuité de l'interaction. C'est le cas du *Pick and Drop* de Rekimoto [12] où des traits jaunes sont projetés sur la table pour mieux percevoir la trajectoire de l'interaction. Ces modèles initiaux sont spécifiés pour une cible donnée, par exemple, le distributeur physique (cible 1 de la Figure 3) ou le poste Internet (cible 2 de la Figure 3).

Les modèles initiaux alimentent le processus de conception. Ce processus se consigne en un ensemble de modèles intermédiaires, dits transitoires. Un consensus émerge pour en identifier trois :

- le lien tâches-concepts qui explicite les concepts manipulés dans chaque tâche. Par exemple, le montant dans le cas d'un retrait bancaire ;
- l'interface abstraite qui structure l'interface en espaces de travail, fixe l'enchaînement entre espaces et identifie le contenu conceptuel de chaque espace. Par exemple, pour le distributeur bancaire, l'interface abstraite de la cible Internet (Figure 1b) est composée de deux espaces de travail, hébergeant respectivement l'identification de l'utilisateur et l'accès aux services ;
- l'interface concrète qui matérialise les espaces de travail, l'enchaînement entre espaces et le contenu de chaque espace en termes d'OIA au sens de Vanderdonckt [15].

L'interface finale est obtenue par codage puis interprétation ou exécution de l'interface concrète. Elle est peuplée d'OIC au sens de Vanderdonckt [15]. Les modèles transitoires et l'interface finale définissent quatre niveaux de réification. A chaque niveau, une référence aux modèles initiaux peut être faite. Plus la référence aux modèles du contexte d'usage est tardive, plus les modèles obtenus sont généraux. La cohérence et donc la continuité de l'interaction devraient en être meilleures.

Le processus consistant à concrétiser progressivement les modèles jusqu'à atteindre l'interface finale est dit « processus de réification » : il s'articule en quatre étapes. L'opération consistant à traduire un modèle en un modèle homologue adapté à une autre cible est dite opération de traduction. La traduction conserve le niveau de

réification : c'est une opération « passerelle ». La traduction est un cas particulier de reciblage dans lequel aucune hypothèse n'est, par contre, émise quant à la conservation du niveau de réification.

Le cadre de référence prévoit que ces opérations (réification, traduction, reciblage) puissent être menées de façon automatique, semi-automatique ou manuelle. La pratique actuelle consiste à produire manuellement, pour chaque cible, les interfaces finales. En l'absence de savoir-faire, la production des modèles transitoires est souvent implicite. Le cadre de référence sert d'outil à la réflexion : il aide les concepteurs à se poser les bonnes questions. Il en est de même pour l'exécution.

Exécution. A l'exécution, des mécanismes sont nécessaires pour (a) la reconnaissance du contexte d'usage (b) le calcul et (c) la mise en oeuvre de la réaction. Ces mécanismes peuvent être embarqués dans les interfaces finales et/ou fournis par une infrastructure d'exécution dédiée à l'adaptation (Figure 3, lettres R pour Reconnaissance, C pour Calcul et M pour Mise en oeuvre). Les mécanismes s'appuient sur les éventuels modèles d'évolution et de transition spécifiés. Dans le cas d'interfaces distribuées, la communication entre composants peut, de la même manière, être fournie par l'infrastructure sous-jacente ou être encapsulée dans les interfaces elles-mêmes. Il en va de même pour le maintien des modèles observés, c'est-à-dire des versions courantes des différents modèles (Figure 3).

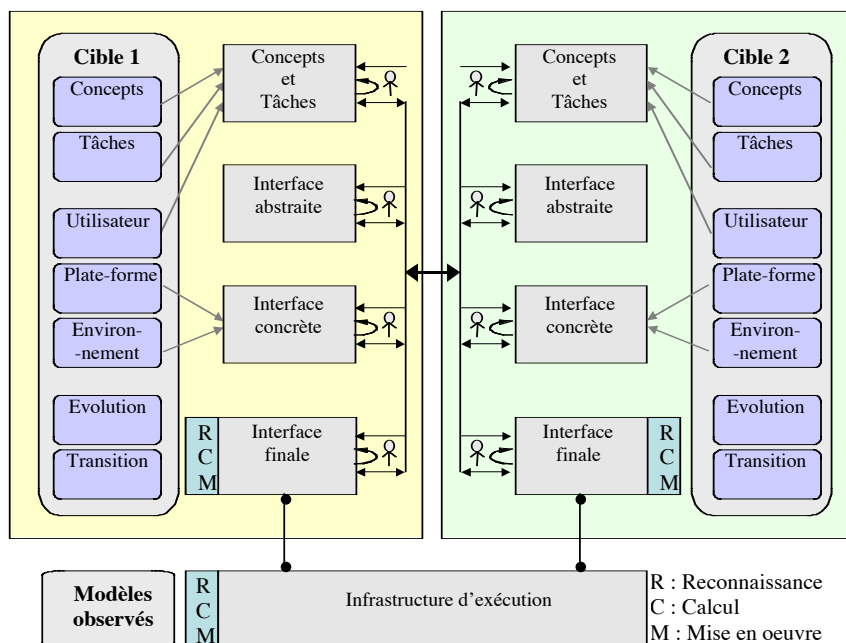


Figure 3 : Le cadre de référence en plasticité . Adapté de [2].

Dans ce cadre de référence, les interacteurs trouvent un double ancrage : en tant que widgets offerts par une boîte à outils, ils font partie intégrante du modèle de la plateforme. En tant que constituants des IHM, ils peuplent les interfaces finales. Notre approche met à profit les acquis en interacteurs et plasticité pour la conception et l'exécution de *comets*.

Notre approche

Notre approche marie les deux grains d'analyse, objets d'interaction (interacteurs) et systèmes interactifs (plasticité), en considérant l'objet d'interaction comme un mini-système interactif. La *comet* devient un système interactif multi-cible mettant à profit les avancées en interacteurs et plasticité. Les sections suivantes présentent les *comets* sous l'angle de la conception puis de l'exécution.

LES COMETS SOUS L'ANGLE DE LA CONCEPTION

Cette section propose une définition et une modélisation des *comets*. Leur architecture logicielle ainsi que celle des systèmes interactifs résultants sont examinées.

Définition

Dans la lignée de Thevenin [14], une *comet* est un interacteur multi-cible, c'est-à-dire une entité qui (a) assure une unité d'interaction entre l'utilisateur et le système, (b) bénéficie de représentations multiples, chacune adaptée à un contexte d'usage donné et garante de propriétés d'utilisabilité, (c) est douée de faculté d'adaptation au contexte d'usage.

Une *comet* sera qualifiée de plastique pour un ensemble de propriétés d'utilisabilité \mathcal{P} et un ensemble de contextes donnés \mathcal{C} si toute propriété de \mathcal{P} est satisfaite sur tout contexte de \mathcal{C} .

Modélisation

Conformément à notre approche, une *comet* est modélisée comme un système interactif (Figure 4) :

- structuré en quatre niveaux de réification : liens concepts-tâches (désormais appelés abstractions), interfaces abstraites, concrètes et finales ;
- doué de capacité d'introspection à tout niveau de réification. Chaque modèle transitoire et final est capable de publier son domaine de validité, à savoir les modèles initiaux supposés ;
- polymorphe à, au moins, un niveau de réification ;
- dont les interfaces finales sont dotées d'une qualité de service. Cette qualité de service exprime (a) le coût de mise en oeuvre de l'interface (les ressources matérielles et logicielles de la plate-forme, les requis environnementaux et humains – ces coûts correspondent exactement aux modèles initiaux relatifs au contexte d'usage de l'interface finale) (b) les pro-

priétés d'utilisabilité garanties par l'interface (c) les effets de bord potentiels de l'interface sur le contexte d'usage (par exemple, l'augmentation du niveau sonore). Ces effets de bord sont désormais banalisés en la propriété de viscosité sur le contexte d'usage.

La figure 4 propose une modélisation UML des *comets*. Elle introduit la classe *ComposantComet* en tant que partie introspective de la *comet* : elle connaît et sait exporter ses modèles initiaux. Une *comet* est composée d'abstractions, d'interfaces abstraites (IU abstraites), d'interfaces concrètes (IU concrètes) et d'interfaces finales (IU finales). Ces composants sont des *ComposantComets* se correspondant mutuellement par réification et inversement abstraction. La *comet* publie sa plasticité pour un ensemble de contextes et de propriétés donnés : c'est un booléen calculé sur les propriétés et les *ComposantComets* des IU finales. L'IU concrète connaît son style (par exemple, le style bouton pour la Figure 1b) et peut dire si son usage est typique dans un contexte donné. L'usage est un réel compris entre 0 et 1.

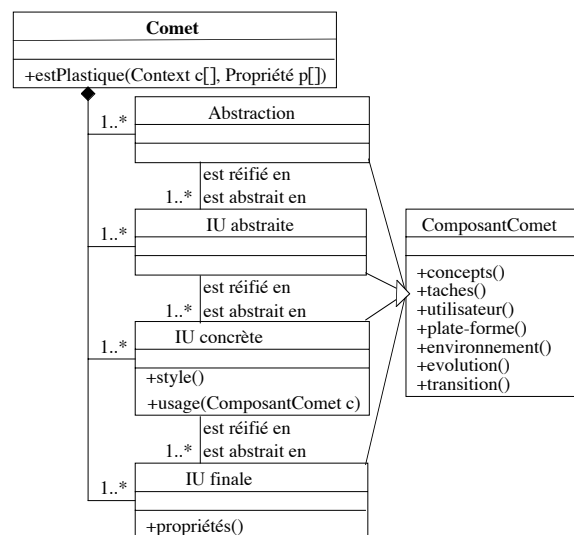


Figure 4 : Diagramme de classes UML d'une *comet*.

La section suivante examine les *comets* sous l'angle de l'architecture logicielle.

Architecture logicielle

Dans la lignée du modèle PAC (Presentation Abstraction Control) [3], nous proposons le modèle rePAC (recursive PAC) peuplant récursivement la facette P d'un agent PAC. La facette P devient un agent PAC dans lequel (Figure 5) :

- l'abstraction A_p se réfère à l'abstraction de la présentation. Elle maintient l'état de l'interaction. Cette facette est déterminante pour des migrations d'IHM à chaud, c'est-à-dire des migrations conservant

l'état de l'interaction. Supposons que l'utilisateur ait choisi sur PDA la consultation de ses comptes bancaires. Si l'IHM est transférée à *chaud* sur PC, alors ce même service sera sélectionné sur PC. C'est cette facette A_p qui maintient l'opération choisie ;

- la facette contrôle assure la gestion de l'adaptation : réception ou diffusion du nouveau contexte d'usage, des adaptations envisageables, de l'adaptation à mettre en oeuvre (réception) ou mise en oeuvre (diffusion). Le sens de la communication (réception versus diffusion) dépend de la politique adoptée. Les politiques sont examinées dans la section suivante. Pour le calcul de l'adaptation, c'est cette facette contrôle C_p qui gère les capacités de polymorphisme de la *comet*. Ces capacités peuvent, par exemple, être maintenues sous la forme d'une liste mémorisant, à chaque niveau de réification (abstraction (A), IU abstraite (IUA), IU concrète (IUC), IU finale (IUF)), les différentes alternatives possibles. Par exemple :

```
((A1 (IUA11 (IUC111 (IUF1111 ... IUF111m))
...
(IUC11n (IUF11n1 ... IUF11no)))
...
(IUA1p (IUC1p1 (IUF1p11 ... IUF1p1q))
...
(IUC1pr (IUF1pr1 ... IUF1prs))))
...
(At ... ))
```

- la facette présentation repère la présentation active de la *comet*. Elle encapsule les widgets traditionnels en les dotant notamment de capacité d'introspection pour qu'ils puissent exporter leurs modèles initiaux. A terme, ces widgets traditionnels seront distribuables sur l'ensemble des plates-formes : c'est la notion de *machine abstraite*.

Dans rePAC, la communication entre facettes reprend les principes de PAC : les facettes A_p et P ne se connaissent pas directement. Elles échangent via la facette C_p .

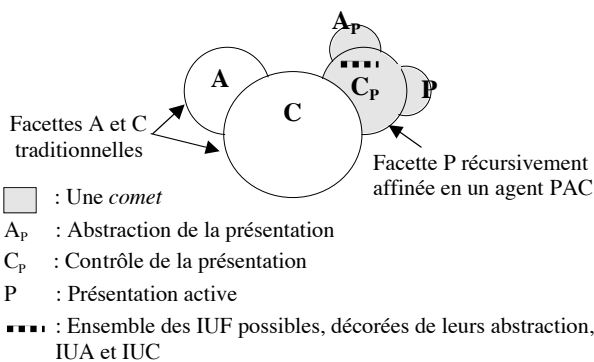


Figure 5 : Architecture logicielle d'une *comet* au sein d'un système interactif : le modèle rePAC.

La figure 6 illustre le modèle rePAC sur la *comet* « Choisir opération bancaire » fédérant les présentations a) b) et c) de la Figure 2. Sur cet exemple, c'est la présentation à base de boutons radios qui est active. Supposons que l'utilisateur sélectionne la consultation de ses comptes : la facette P capte l'événement, le transmet à la facette C_p , qui le propage aux facettes C et A_p . A_p maintient le service sélectionné. Supposons que l'utilisateur retaile alors la fenêtre en largeur. Cet événement est capté par la fenêtre (elle-même une *comet*) et acheminé à ses fils, dont la facette C qui propage l'événement au C_p en charge de l'adaptation. Différentes stratégies sont alors envisageables (elles font l'objet de la section suivante). En cas de pleine autonomie de la *comet*, C_p consulte le modèle d'évolution de la présentation à base de boutons radios. Ce modèle suggère la commutation vers la présentation à base de menu déroulant (Figure 2c). C_p récupère alors auprès de A_p l'état de l'interaction. Il réinitialise la facette P avec cette nouvelle présentation conformément à cet état.

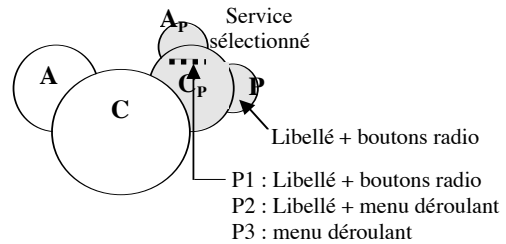


Figure 6 : Illustration du modèle rePAC.

Dans cette lignée, un système interactif est une hiérarchie d'agents rePAC, les contrôles de l'adaptation (facettes C_p) étant reliés entre eux (directement ou par leurs C) pour propager la gestion du contexte d'usage au sein de la hiérarchie. La figure 7 illustre cette hiérarchie sur l'exemple du distributeur bancaire présenté en Figure 1b. Dans cet exemple, la hiérarchie d'agents coïncide avec le modèle des tâches : les tâches abstraites sont incarnées par des espaces de travail concrétisés en des *comets* fenêtres ou canevas. Ces *comets* se coordonnent pour une cohérence ergonomique de l'interface.

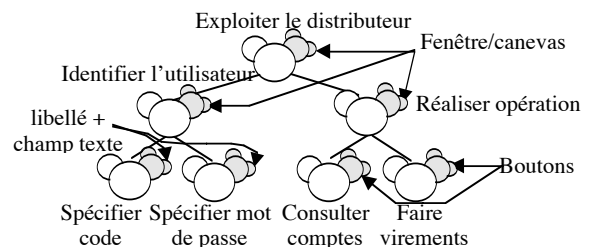


Figure 7 : Illustration de l'architecture logicielle d'un système interactif dans le style rePAC.

La section suivante est consacrée à l'exécution.

LES COMETS SOUS L'ANGLE DE L'EXECUTION

Cette section présente les stratégies et politiques d'adaptation. Les stratégies s'appuient sur une Nouvelle Génération (NG) de boîtes à outils.

Boîtes à outils NG

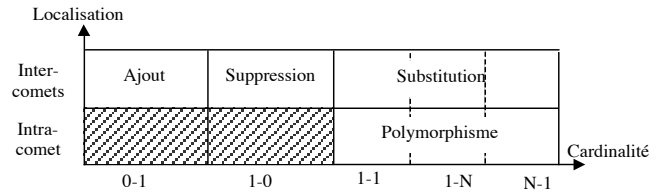
L'idée des boîtes à outils NG est d'augmenter les traditionnelles bibliothèques de mécanismes de recherche multi-critères. Ces mécanismes permettraient, par exemple, d'identifier l'ensemble des *comets* disponibles sur la plate-forme P (par exemple, la classe des PDA), supportant la tâche T (par exemple, la sélection d'un élément parmi N) et douées de polymorphisme au plus faible niveau de réification. D'autres permettraient, au prix d'un affinement de la tâche T en un ensemble de sous-tâches T1...Tn, d'identifier des *comets* offrant les sous-tâches T1...Tn, et donc fonctionnellement complémentaires au regard de la tâche T. La section suivante est consacrée aux stratégies d'adaptation.

Stratégies

Nous identifions quatre classes de stratégies :

- l'adaptation par polymorphisme. C'est une stratégie qui conserve la *comet* mais en change la forme. Ce changement de forme peut s'opérer à tout niveau de réification selon trois cardinalités, 1-1, 1-N, N-1 selon que la forme est remplacée par une autre (cardinalité 1-1), N autres (cardinalité 1-N) ou que N formes sont agrégées en une seule (cardinalité N-1). Typiquement, dans la figure 2 :
 - La commutation des boutons radios en un menu déroulant (Figure 2a à 2b) est un exemple de polymorphisme 1-1 ;
 - L'engloutissement du libellé dans le menu déroulant (Figure 2b à 2c) est un exemple de polymorphisme N-1 ;
 - Inversement, l'externalisation du libellé (Figure 2c à 2b) est un exemple de polymorphisme 1-N.
- l'adaptation par substitution. Par opposition au polymorphisme, cette stratégie ne préserve pas la *comet*. Elle la remplace par un ou N autres. On distingue, de même, trois types de substitution selon leur cardinalité 1-1, 1-N, N-1. Les mêmes exemples que précédemment pourraient être repris pour des *comets* n'offrant pas les capacités de polymorphisme mentionnées ;
- l'adaptation par ajout. Une *comet* est ajoutée au système interactif. C'est le cas typiquement de la représentation multiple d'un même concept, sollicitée pour de l'insistance temporaire ;
- l'adaptation par suppression : c'est la duale de l'ajout. Une *comet* est supprimée parce que l'information n'est plus critique ou que la tâche ne fait plus sens dans le contexte d'usage courant.

La figure 8 présente la carte des stratégies disponibles.



//// : ne font pas sens, les Comets ayant au moins une interface finale

Figure 8 : Revue des stratégies envisageables.

A l'exécution, les stratégies seront dictées par le modèle d'évolution de la *comet* ou examinées selon un ensemble de critères. Par exemple, le maintien ou non maintien de l'abstraction, de l'interface abstraite, de l'interface concrète et des propriétés d'utilisabilité de l'interface finale. Dans l'esprit de QOC (Questions Options Criteria) [9], une notation pourrait être proposée comme aide à la décision et support à la traçabilité. Cette notation permettrait d'exprimer, pour chaque niveau de réification, des requis de maintien, non maintien (augmentation, réduction) ou d'indifférence. L'IU finale serait examinée par le biais de ses propriétés. Chaque exigence serait pondérée d'un poids reflétant son importance.

Les stratégies sont déployées selon des politiques présentées dans la section suivante.

Politiques

Les politiques dépendent de l'autonomie accordée à la *comet* dans sa prise en charge de l'adaptation : reconnaissance de la situation, calcul et mise en oeuvre de la réaction. Pour chacun de ces aspects, un degré d'autonomie est accordé à la *comet*. Ce degré varie de 0 à 1, donnant lieu à quatre classes de politiques :

- la politique *Non concertée externe* correspond à une autonomie nulle : la *comet* est incapable de gérer l'adaptation. Elle doit être prise en charge par un tiers (une autre *comet* ou l'infrastructure d'exécution) ;
- la politique *Non concertée interne* correspond à une autonomie totale : la *comet* assume, en toute indépendance, l'adaptation ;
- les politiques *Concertées* correspondent à une autonomie partielle : une collaboration s'établit entre la *comet* et un tiers (une autre *comet* ou l'infrastructure d'exécution) pour la prise en charge de l'adaptation. Tandis qu'une version optimiste autorisera la *comet* à appliquer la décision sans accord préalable du tiers, une version pessimiste requerra cet accord avant toute application. Une version optimiste s'expose à devoir, en cas d'erreur, annuler les mesures mises en oeuvre.

Le choix de la politique se fera au regard de critères tels que la performance par exemple. Seule la politique non concertée interne est aujourd'hui mise en oeuvre.

MISE EN OEUVRE

Les *comets* sont aujourd'hui implémentées en C++/OpenGL 1.4. Une boîte à outils NG est en cours de développement pour guider le concepteur dans son assemblage de *comets*. Les systèmes interactifs respectent le modèle rePAC. La stratégie de polymorphisme et la politique non concertée interne sont implémentées. Les autres stratégies et politiques figurent parmi nos perspectives.

CONCLUSION ET PERSPECTIVES

En conclusion, rappelons que cet article présente une révision des widgets consistant à les doter de faculté d'adaptation au contexte d'usage. Basée sur une revue critique de l'état de l'art, cette révision s'articule autour de la notion de *comet*. Une définition, modélisation et architecture logicielle en sont proposées. Elles permettent la composition de *comets* au sein de systèmes interactifs. L'exécution suppose le choix de stratégies et politiques d'adaptation. Des taxonomies et grilles d'analyse en sont proposées.

Les perspectives à ce travail sont nombreuses. Il convient tout d'abord de poursuivre la mise en oeuvre en l'étendant aux stratégies et politiques non encore implémentées. Il conviendra alors de modéliser plus finement les modèles initiaux, en particulier le contexte d'usage pour une appréciation plus juste des coûts, propriétés et effets de bord des *comets*. Des efforts de catégorisation et normalisation seront nécessaires, notamment en termes de tâches (identifier, nommer et capitaliser les tâches d'utilité publique) pour le déploiement de boîtes à outils NG. Ces efforts initiés par Vanderdonckt [15] sont abstraits par Thevenin [14] qui distingue les interacteurs de présentation et de navigation. Il s'agira de poursuivre ce travail et d'identifier notamment la *bonne* granularité des *comets*. A plus long terme, nous prévoyons l'étude de la continuité de l'interaction pour préconiser des patrons d'adaptation. Cette réflexion sera initiée sur les interfaces graphiques puis étendue aux autres modalités, notamment le son et le tactile. Les odeurs font partie de nos perspectives mais à bien plus long terme ...

REMERCIEMENTS

Ce travail a bénéficié du soutien du projet européen IST CAMELEON (IST-2000-30104). Nous remercions particulièrement Jean Vanderdonckt, membre du projet, pour le concours apporté à l'étude. Nous remercions, par ailleurs, Christophe Lachenal et Nicolas Barralon de l'équipe IIHM pour leurs remarques constructives.

BIBLIOGRAPHIE

1. Calvary, G., Coutaz, J. *Plasticité des Interfaces : une nécessité !*, Actes des deuxièmes Assises nationales du GDR I3, Cépaduès Editions, J. Le Maître (Ed), Nancy, Décembre 2002, pp 247-26.
2. 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.
3. Coutaz, J. *PAC, an Object Oriented Model for Dialog Design*, In Interact'87, 1987, pp 431-436.
4. Crease, M. *A Toolkit Of Resource-Sensitive, Multimodal Widgets*. PhD Thesis, Department of Computing Science, University of Glasgow, Dec 2001.
5. Dâassi, O., Calvary, G., Coutaz, J. *Les Interacteurs en Plasticité*, Annexe des actes d'IHM'02, pp 25-28.
6. Demeure, A., Calvary, G. *Le Modèle d'Evolution en Plasticité des Interfaces : Apport des Graphes Conceptuels*, Actes d'IHM 2003.
7. Duke, D.J., Harrisson, M.D. *Abstract Interaction Objects*, Eurographics 93, Spain, 1993.
8. Faconti, G., Paterno, F. *An approach to the formal specification of the components of an interaction*. In C. Vandoni and D. Duce, editors, Eurographics 90, pp 481-494, Noth-Holland, 1990.
9. MacLean, A., Young, R.M., Bellotti, V., Moran, T. *Questions, Options and Criteria : Elements of Design Space Analysis*. In Human-Computer Interaction, 6 (3 & 4), 1991, pp 201-250.
10. Markopoulos, P. *Interactors: formal architectural models of user interface software*. In Kent, A., and Williams, J.G., (Eds.) Encyclopedia of Microcomputers, Volume 27 (Supplement 6), Marcel Dekker, New York, 2001, pp 203-235.
11. Mori, G., Paternò, F., Santoro, C. *Tool Support for Designing Nomadic Applications*, In Proceedings of IUI 2003 -Miami, Florida, January 12-15, 2003.
12. Rekimoto, J. *Pick and Drop: A Direct Manipulation Technique for Multiple Computer Environments*. In Procs of UIST97, ACM Press, 1997, pp. 31-39.
13. Thevenin D., Coutaz J. *Plasticity of User Interfaces: Framework and Research Agenda*, Proc. Interact'99, Edinburgh, A. Sasse et C. Johnson (dir.), IFIP IOS Press, pp 110-117, 1999.
14. Thevenin, D. *Adaptation en Interaction Homme-Machine : Cas de la plasticité*. Thèse de doctorat de l'Université Joseph Fourier, Décembre 2001.
15. Vanderdonckt, J. *Conception assistée de la présentation d'une interface homme-machine ergonomique pour une application de gestion hautement interactive*. Thèse des Facultés Universitaires Notre-Dame de la Paix, Spécialité Informatique, Juillet 1997.