

# Des widgets aux comets pour la Plasticité des Systèmes Interactifs

## *From widgets to comets for supporting software plasticity*

Gaëlle CALVARY, Oifa DAASSI, Joëlle COUTAZ, Alexandre DEMEURE

Laboratoire CLIPS-IMAG, BP 53, 38041 Grenoble cedex 9, France  
Gaelle.Calvary@imag.fr, Joelle.Coutaz@imag.fr

**Résumé.** Cet article traite de l'adaptation des systèmes interactifs à leur contexte d'usage, le contexte d'usage étant défini en termes d'utilisateur, de plate-forme et d'environnement d'interaction. Plus précisément, c'est la *plasticité des systèmes interactifs* qui est au cœur de l'étude, c'est-à-dire la capacité des systèmes interactifs à s'adapter ou être adaptés à leur contexte d'usage dans le respect de leur utilisabilité. L'article propose une nouvelle génération de widgets, les « comets » (Context Mouldable widgETs), pour la construction de systèmes interactifs plastiques. Une comet est, par définition, un widget autodéscriptif qui rend observables ses propriétés fonctionnelles et non fonctionnelles. Une comet peut être polymorphe en termes d'incarnation. Le polymorphisme peut s'exercer à différents niveaux d'abstraction : concepts et tâches, interfaces abstraite, concrète et finale. Chaque niveau se décrit d'un point de vue fonctionnel et non fonctionnel. Une comet peut être douée d'auto-adaptation, l'adaptation pouvant mettre à profit l'éventuel polymorphisme de la comet. L'article couvre la conception et l'exécution des comets. Il en souligne l'originalité par rapport à l'état de l'art.

**Mots-clés.** Adaptation, contexte d'usage, plate-forme, environnement, utilisateur, plasticité, interacteur, widget, comet, Compact.

**Abstract.** This paper addresses software adaptation to context of use. It proposes a new generation of widgets for supporting software plasticity. Plasticity refers to the ability of an interactive system to withstand variations of context of use while preserving its quality in use. A comet is a self-descriptive widget that is able to self-adapt to some context of use, or that can be adapted by a tier-component to the context of use, or that can be dynamically discarded (versus recruited) when it is unable (versus able) to cover the current context of use. To do so, a comet publishes the quality in use it guarantees, the user tasks and the domain concepts that it is able to support, as well as the extent to which it supports adaptation. Adaptation may rely on polymorphism. Polymorphism may occur at any level of abstraction.

**Keywords.** Adaptation, context of use, platform, environment, user, plasticity, interactor, widget, comet, Compact.

## 1 Introduction

Avec la diversité des plates-formes, leur assemblage opportuniste et leur usage en conditions mobiles, l'adaptation des systèmes interactifs au contexte d'usage est étudiée comme réponse à un contexte d'usage désormais *varié* et *variable*. 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 d'une 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.

Jusqu'aux travaux de Crease (Crease, 2001), l'adaptation était étudiée de façon globale aux systèmes interactifs. En particulier, ArtStudio (Thevenin, 2001), Teresa (Mori et al., 2003) et leurs précurseurs Mastermind, Tadeus, FUSE ou TRIDENT (Vanderdonckt, 1997) produisaient des interfaces Homme-Machine (IHM) par assemblage d'objets d'interaction *rigides*, c'est-à-dire non doués de faculté d'auto-adaptation. La priorité était mise sur les aspects fonctionnels, ne considérant des objets d'interaction que la tâche (implicite) qu'ils supportaient.

Les *comets* (Context Mouldable widgETs) sont une génération de widgets, spécialement façonnés pour la *plasticité*, c'est-à-dire l'adaptation des systèmes interactifs à leur contexte d'usage dans le respect de leur utilisabilité (Thevenin et Coutaz, 1999). Cet article prolonge la publication faite à IHM'03 (Daassi et al., 2003). Après une première section consacrée à deux cas d'étude qui motivent et illustrent la notion de comète, l'article propose un état de l'art (section 3) dont découlent les lignes de force des comètes. Leur conception puis exécution font l'objet des sections 4 et 5. Leur mise en œuvre est présentée en section 6. Elle donne lieu à de nombreuses perspectives (section 7).

## 2 Cas d'étude et esquisse des comètes

Cette section présente deux cas d'étude de natures et finalités diverses. Le premier est un cas d'étude réel : il motive les comètes par les incohérences ergonomiques induites par des ingénieries au cas par cas. Le second est une application « jouet » à finalité d'illustration de l'article.

### 2.1 Le distributeur bancaire

Bob souhaite 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).

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 le 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 dans la mesure où les pages d’écran se succèdent au fil de l’interaction. Aussi, des traits blancs (encadrés pour le retrait rapide de 40 € en Figure 1a) 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).

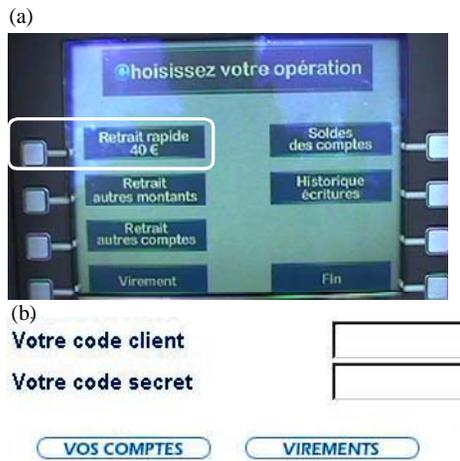


Figure 1. Opérations bancaires sur distributeur (cas a) vs Internet (cas b).

Si, pour des tâches simples comme la consultation bancaire, l'utilisateur s'accommode d'incohérences ergonomiques, ces incohérences peuvent être pénalisantes pour des tâches plus complexes ou des utilisateurs stressés. La voie ici explorée est celle de l'abstraction et de la metadescription. Il s'agit :

- d'abstraire des IHM la tâche utilisateur supportée. Par exemple, voir en Figure 1a), la gestion d'un compte bancaire ;
- de décrire chaque IHM d'un point de vue fonctionnel (tâche utilisateur supportée et concepts du domaine manipulés – par exemple, gérer son compte bancaire pour la Figure 1a) et non fonctionnel (par exemple, la longueur de la trajectoire d'interaction requise pour l'accomplissement de la tâche) ;
- et d'établir des classes d'équivalence fonctionnelle et/ou non fonctionnelle entre IHM. Les IHM de la figure 2 offrent typiquement la même tâche utilisateur mais diffèrent dans leur coût en termes de surface d'affichage et de longueur de trajectoire d'interaction. Tandis que la version 2c) sera appropriée pour des surfaces d'affichage restreintes, la version 2a) sera un bon candidat si l'interaction doit être réalisée en un minimum d'actions physiques.



Figure 2. Trois incarnations possibles de la même tâche utilisateur.

Une même comète pourra embarquer les trois formes a), b) et c) de la Figure 2, devenant ainsi *polymorphe* en présentation concrète. Elle pourra être dotée de mécanismes d'adaptation, la rendant *auto-adaptative*. Ces notions seront développées en section 4.2.

La section suivante présente un cas d'étude jouet qui servira de support à l'illustration de l'article. Il couvre, de façon complète, les leviers de la plasticité.

### 2.2 La double horloge

Pour Bob, un jeune américain, se déplaçant épisodiquement à Paris, la montre idéale est une montre à double cadrans affichant conjointement l'heure à Paris et New-York. Plusieurs variantes existent dans le commerce : cadrans adjacents, heures emboîtées. Bob privilégie la lisibilité et opte pour les cadrans adjacents. A l'usage, il se rend compte que cette solution n'est pas idéale : il lui arrive d'oublier sa montre ou de la retrouver sans pile. PlasticClock est une alternative numérique. Elle s'exécute sur PC et/ou assistant personnel (PDA pour Personal Digital Assistant), se remodelant au changement de plate-forme.

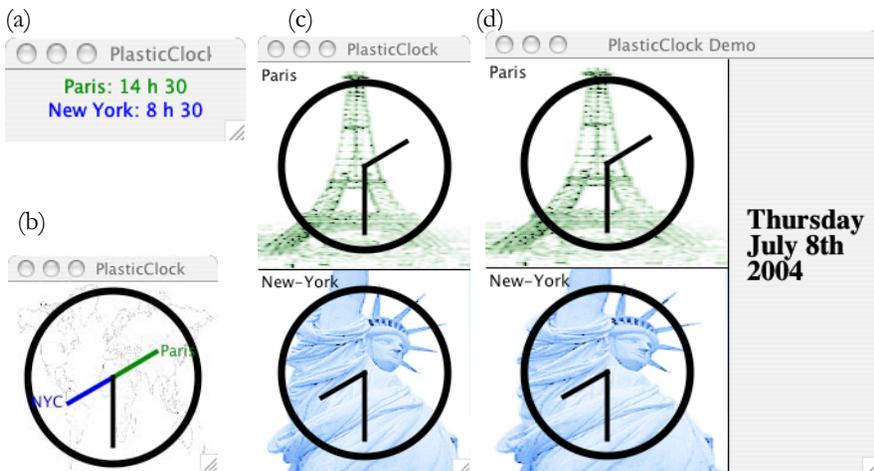
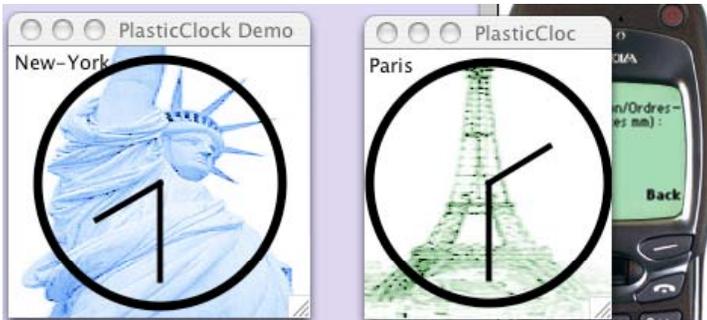


Figure 3. Les adaptations de PlasticClock en version centralisée. Les heures à Paris et New-York sont toujours affichées tandis que la date n'apparaît que lorsque la surface d'affichage le permet. Selon la place disponible, les heures sont représentées par un texte (a), un cadran commun (b) ou spécifique (c).

PlasticClock s'adapte à la taille d'écran. Dans sa version minimale, elle affiche les heures au format texte (Figure 3a). En version maximale, elle opte pour deux cadrans et rajoute la date du jour (Figure 3d). Entre ces deux extrêmes, les heures sont, dans un premier temps, regroupées en un même cadran (Figure 3b) puis scindées en deux dès lors que la taille d'écran le permet (Figure 3c). Ce sont des exemples de *remodelage*.

Supposons maintenant que Bob veuille disposer de l'heure de Paris sur son téléphone. Il lui suffira demain (non implémenté aujourd'hui) d'approcher son téléphone du PC. Une icône apparaîtra sur le bureau, matérialisant cette nouvelle plate-forme (Figure 4). Par manipulation directe, Bob sélectionnera l'heure de Paris (le cadran ou l'aiguille), la glissera sur le bureau puis la déposera sur l'icône du téléphone. La migration s'initiera et l'heure de Paris apparaîtra (peut-être remodelée) sur le téléphone. C'est un exemple de *redistribution*.



**Figure 4.** *PlasticClock : du centralisé au distribué. Bob détache l'heure de Paris, la glisse sur l'icône matérialisant la plate-forme cible : l'heure de Paris migre sur le téléphone. C'est un exemple de système interactif redistribuable.*

PlasticClock se veut un démonstrateur de la plasticité. Il en illustre les deux leviers, à savoir remodelage (Figure 3) et redistribution (Figure 4). Il étend en cela FlexClock (Grolaux et al., 2001) limité au seul remodelage.

La section suivante présente un état de l'art des travaux pertinents pour la conception et l'exécution des comets. Il est illustré sur PlasticClock.

### 3 Etat de l'art

Cette section examine l'état de l'art en plasticité et interacteurs en vue de la conception et de l'exécution des comets. Elle permet d'énoncer notre approche et d'en établir les hypothèses de travail.

#### 3.1 Plasticité

Les deux leviers de la plasticité, remodelage et redistribution (illustrés dans la section 2.2), sont couverts, du point de vue de l'ingénierie, par un même cadre de référence (Calvary et al., 2003). Ce cadre de référence unifie la conception et l'exécution d'IHM plastiques en une approche basée modèles (Ingénierie Dirigée par les Modèles aussi dite MDE ou MDA pour Model Driven Engineering ou Architecture). Conception et exécution sont ici examinées.

#### *Conception*

La conception s'articule autour d'un ensemble de modèles (dits *initiaux*) spécifiés par le concepteur : les modèles du domaine (concepts et tâches utilisateur),

du contexte d'usage (utilisateur, plate-forme, environnement), de l'adaptation et de la qualité d'usage à préserver. Nous développons ici les modèles d'adaptation et de qualité.

Le modèle d'adaptation spécifie les règles d'évolution et de transition à appliquer en cas de changement de contexte d'usage. Le modèle d'évolution (Demeure et Calvary, 2003) spécifie la réaction à mettre en oeuvre en cas de changement de contexte d'usage : par exemple, migrer l'heure de Paris sur le téléphone lorsque ce dernier s'approche de la plate-forme courante. Le modèle de transition complète cette spécification par un accompagnement de l'utilisateur dans le changement : par exemple, la migration de l'heure sur une autre plate-forme peut être signalée par la déformation de son cadran (un étirement vers la destination cible, par exemple). Dans *Pick and Drop* (Rekimoto, 1997), les lignes jaunes projetées sur la table relèvent de cette vocation de transition : elles aident l'utilisateur à percevoir la trajectoire d'interaction pour une meilleure continuité de l'interaction. La continuité de l'interaction peut être un critère qualité explicitement mentionné dans la qualité d'usage attendue.

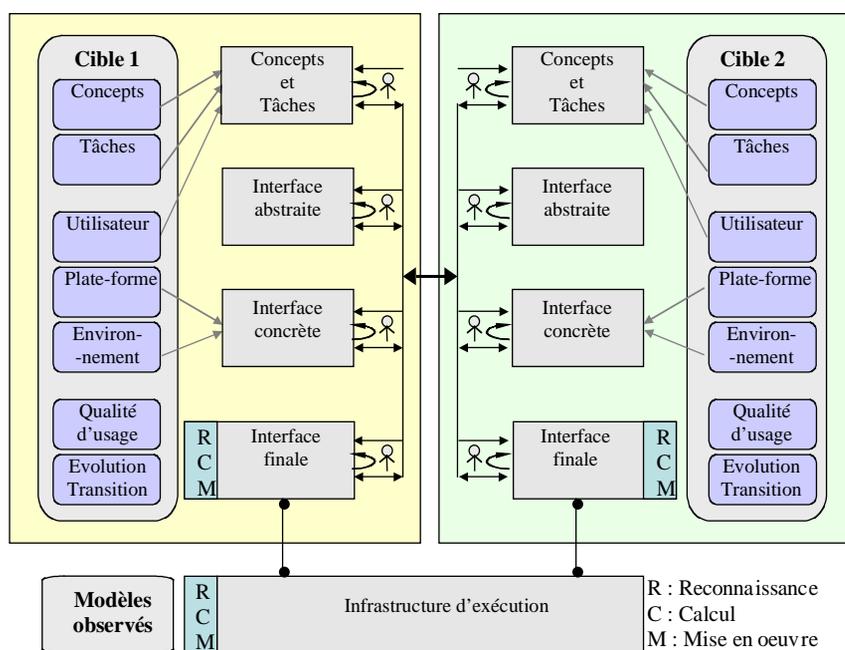
La qualité d'usage spécifie les propriétés à préserver lors de l'adaptation, c'est-à-dire l'utilisabilité à garantir pour que le système interactif soit dit plastique. Le terme de « qualité d'usage » est ici emprunté à l'ISO (2003a). Il est préféré au terme d'« utilisabilité » pour ne pas oublier la préservation des requis fonctionnels (c'est-à-dire l'engagement à satisfaire une certaine tâche utilisateur). La spécification de la qualité d'usage doit s'appuyer sur un cadre de référence, par exemple, celui de l'ISO à base de caractéristiques et métriques (2003b). D'autres référentiels existent, notamment pour l'utilisabilité : par exemple, par ordre chronologique, Shackel (1991), Abowd et al. (1992), Bastien et Scapin (1993), Dix et al. (1993), Nielsen (1994), Preece et al. (1994), les propriétés de l'IFIP (1996), Schneiderman (1997), Constantine et Lockwood (1999) ou encore Van Welie et al. (1999). D'autres travaux spécifiques ont été menés pour des domaines particuliers tels que le web (Montero et al., 2004) ou les applications militaires. Récemment Lopez-Jacquero et al. (2004) ont proposé un référentiel spécifique à l'adaptation. QUIM (Quality in Use Integrated Map) (Seffah et al., 2001) se veut un modèle unificateur à base de données, métriques, critères et facteurs. Ce référentiel s'appuie sur les critères ergonomiques de Bastien et Scapin (1993) qu'il revisite pour notamment inclure la continuité de l'interaction. En l'absence de consensus aujourd'hui quant au cadre de référence, il semble sage de laisser ouvert le choix du référentiel, mais d'imposer l'adossement de la définition et de sa mesure sur un référentiel donné. Le choix du référentiel n'est pas l'objet de l'article.

Les modèles initiaux alimentent le processus de conception. Ce processus se consigne en un ensemble de modèles intermédiaires (dits *transitoires*) et *finaux* (figure 5) :

- Le modèle tâches-concepts explicite les tâches utilisateur et les concepts qu'elles manipulent. Par exemple, en version petit et moyen écran (Figures 3a, b et c), la tâche utilisateur se limite à la consultation des deux heures. En version grand écran (Figure 3d), elle inclut la consultation de la date ;
- L'interface abstraite structure l'IHM en espaces de travail, fixe l'enchaînement entre espaces et identifie le contenu conceptuel de chaque espace. Par exemple, en version petit écran (Figures 3a et b), les IHM se limitent à un espace de travail unique rendant observables les deux heures. La version moyen écran (Figure 3c) est structurée en deux espaces de travail, accessibles à tout instant. Ils rendent observables les deux heures. La version grand écran (Figure 3d) se décompose en deux espaces de travail (un pour les heures, un pour les dates), l'espace de travail

dédié aux heures étant récursivement décomposé en deux espaces de travail (un pour Paris, un pour New-York). En pratique, l'interface abstraite est une transformation du modèle tâches-concepts par l'application de règles, par exemple : à chaque tâche est associé un espace de travail, le contenu n'étant ni plus ni moins que les concepts manipulés dans la tâche. L'enchaînement entre espaces se déduit des relations entre tâches ;

- L'interface concrète matérialise les espaces de travail, l'enchaînement entre espaces et le contenu de chaque espace en interacteurs (aussi dits objets d'interaction ou widgets). En graphique, les espaces de travail deviennent fenêtres ou canevas ; la navigation devient nulle (Figure 3) ou s'incarne en listes déroulantes, boutons radio (Figure 2), boutons d'activation, onglets ou liens hypertexte ;
- L'interface finale est obtenue par codage puis interprétation ou exécution de l'interface concrète.



**Figure 5.** Le cadre de référence en plasticité. Adapté de (Calvary et al., 2003).

Les modèles transitoires et l'interface finale définissent quatre niveaux de *réification* (ou concrétisation). 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.

Ce cadre de référence permet de raisonner sur l'adaptation des systèmes interactifs et de caractériser les approches existantes (Calvary et al., 2003). Les travaux sur les systèmes interactifs sensibles au contexte (par exemple, (Stephanidis et Akoumianakis, 2001 ; Grundy et Yang, 2003)) pourraient être caractérisés sur cette base. Ce n'est pas l'objet de l'article.

La figure 6 présente, à titre illustratif, deux instanciations fréquentes du cadre de référence : en a), la dépendance au contexte d'usage est retardée au niveau de l'interface concrète, tandis qu'en b), des hypothèses sont émises dès le modèle tâches-concepts rendant chaque modèle dépendant de son contexte d'usage. En pratique, la dépendance peut être affinée en termes de :

- Modalité d'interaction (graphique, sonore, tactile, etc.) : une telle dépendance est classique au niveau de l'interface concrète, mais peut apparaître dès le modèle des tâches. Typiquement, « sélectionner un objet puis le détruire » correspond implicitement à une interaction graphique. En vocal et déictique, la procédure serait plutôt de spécifier l'action (prononcer « détruire ») puis désigner l'objet en prononçant « cet objet » ;
- Interacteurs (les objets d'interaction ou widgets disponibles dans la boîte à outils) : la dépendance est manifeste au niveau de l'interface concrète. En réalité, elle apparaît dès le modèle des tâches, le niveau de décomposition du modèle dépendant implicitement des interacteurs disponibles. Typiquement, si PlasticClock est un interacteur d'utilité publique, alors la tâche « consulter deux heures » ne sera pas décomposée plus avant ;
- Environnement d'exécution : ce dernier cran permet d'identifier des rendus dépendants de l'environnement tel qu'un navigateur Internet par exemple.

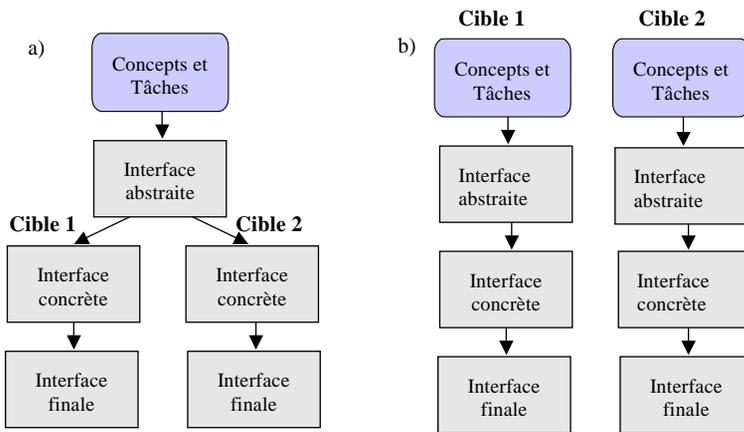


Figure 6. Deux instanciations du cadre de référence.

Le cadre de référence prévoit que les opérations de réification (ou concrétisation), abstraction (réification inverse) et reciblage (changement de cible) puissent être menées de façon automatique, semi-automatique ou manuelle. La pratique actuelle consiste à produire manuellement, pour chaque cible, les interfaces concrètes et/ou finales. L'avenir pourrait être la composition dynamique d'IHM ainsi préfabriquées. Ces IHM seraient métadécrites à tout niveau d'abstraction et capitalisées dans une base de données (ou annuaire de services) (Figure 5).

**Exécution**

L'adaptation au contexte d'usage est un processus en trois étapes comprenant : la reconnaissance du contexte d'usage, le calcul puis la mise en œuvre de la réaction (Calvary et al., 2001).

La reconnaissance de la situation comprend :

- La capture du contexte d'usage : par exemple, dans PlasticClock, la taille de la fenêtre (ex : 120x60 pixels) ;
- La détection du changement de contexte (ex : passage de 120x60 à 140x80 pixels) ;
- L'identification du changement (ex : passage d'une *petite* à une *grande* fenêtre). C'est cette identification qui va déclencher une éventuelle réaction. On distingue trois types de déclencheurs : l'entrée dans un contexte, la sortie d'un contexte et la présence dans un contexte (Schmidt, 1999). Les déclencheurs sont combinables par conjonction et disjonction. Ainsi "sortie(C1) et entrée(C2)" est un déclencheur exprimant la transition du contexte C1 au contexte C2.

Le calcul de la réaction couvre :

- Le recensement des réactions possibles. Deux approches sont envisageables : intervenir à la source en modifiant le contexte d'usage (le rétablir par exemple en allumant la lumière lorsque la nuit tombe) ou adapter le système interactif au nouveau contexte. Les leviers illustrés en section 2 (remodelage et redistribution) sont alors à examiner ;
- Le choix d'une solution parmi les candidates. La qualité d'usage attendue sera considérée. Typiquement, si le temps d'adaptation est une donnée importante, une solution à base d'IHM préfabriquées sera préférée à une génération à la volée, qui produirait certes du « sur mesure » mais dans des temps de calcul incompatibles de la qualité attendue.

La mise en œuvre consiste à :

- Préparer la réaction. Ce prologue comprend la gestion de la tâche en cours (terminaison, suspension ou avortement), la sauvegarde du contexte d'interaction et la production de l'IHM cible (par réification de modèles existants, recyclage ou composition d'IHM préfabriquées) si cette IHM n'a pas encore été élaborée ;
- Exécuter la réaction (par exemple, commuter de la Figure 3a à 3b) ;
- Terminer la réaction. Cet épilogue comprend la restauration du contexte d'interaction sauvegardé lors du prologue.

Les mécanismes de reconnaissance de contexte d'usage (R), de calcul (C) et de mise en œuvre (M) de la réaction peuvent être embarqués dans les systèmes interactifs eux-mêmes et/ou offerts par une infrastructure d'exécution (Figure 5). On parlera d'*adaptation interne* (respectivement *externe*) (traductions des *close* et *open adaptations* de Oreizy et al. (1999)) lorsque l'adaptation est prise en charge par le système interactif lui-même (respectivement, un tiers, à savoir un autre système interactif ou l'infrastructure d'exécution). En cas de collaboration entre les systèmes interactifs et/ou l'infrastructure d'exécution, on parlera d'*adaptation mixte*. PlasticClock est un exemple d'adaptation interne : les mécanismes d'adaptation sont embarqués dans le système interactif lui-même.

Quelle que soit la localisation de l'adaptation (interne, externe ou mixte), la reconfiguration dynamique s'avère mieux supportée par une approche de type composant-connecteur (Oreizy et al., 1999) (De Palma et al., 1999) (Garlan et al., 2001). Les composants réflexifs (i.e. doués de capacités d'analyse et d'adaptation de leur propre comportement) sont à la base des adaptations internes tandis que l'adaptation externe s'appuie plus naturellement sur des composants introspectifs (i.e. métadécrits (Englander, 2001 – voir le chapitre 9 sur l'introspection)).

La section suivante examine, sous l'angle de la plasticité, l'état de l'art en interacteurs.

### 3.2 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 (Faconti et al., 1990). Dès 1993, Duke et Harrison 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 (Duke et Harrison, 1993). Dès lors, le terme d'« interacteur » semble préférable à la notion de widget, consacrée aux interfaces graphiques. Depuis, plusieurs contributions à dominante ingénierie (par opposition à l'approche analytique de Duke et Harrison) ont été apportées. Nous retenons de cet état de l'art :

- La distinction entre interacteurs abstraits et concrets (Morin, 1990). La première implémentation en est donnée par Vanderdonck et Bodardt (1993). Elle sépare la fonction de l'interacteur (la tâche utilisateur qu'il offre, par exemple la sélection d'une option parmi N) de son style de présentation (par exemple, les boutons radio ou listes déroulantes). Cette distinction rejoint les niveaux d'abstraction du cadre de référence en plasticité : la tâche est exprimée au niveau tâches-concepts ; le style d'interaction intervient dans l'interface concrète ;
- Des tentatives d'unification du fonctionnel et du non fonctionnel pour raisonner sur l'ergonomie du système interactif (De Abreu-Cybis, 1994) (Farenc et al., 1995). Les travaux restent rares, ce qui est sans doute dû à un manque d'opérationnalisation des critères et métriques en matière de qualité d'usage ;
- Le façonnage des interacteurs pour l'adaptation au contexte d'usage. Dey et al. (1999) s'intéressent, en particulier, à la capture du contexte d'usage et l'architecture permettant son abstraction (la Reconnaissance du contexte d'usage sur la Figure 5). Crease (2001) conçoit des interacteurs polymorphes au niveau concret (multimodaux, son et graphique) qu'il dote de mécanismes d'adaptation ;
- La composition d'interacteurs. Dans la lignée de Duke et Harrison (1993), Markopoulos (2001) 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.

Les comets rassemblent les acquis en plasticité et interacteurs. Notre approche ainsi que nos hypothèses de travail font l'objet de la section suivante.

### 3.3 Approche et hypothèses

Notre approche marie les deux grains d'analyse, interacteurs et systèmes interactifs (plasticité), considérant l'interacteur comme un mini-système interactif. Cette hypothèse du multi-échelle semble raisonnable, l'état de l'art des interacteurs confirmant :

- L'existence de niveaux d'abstraction (abstrait et concret) ;
- La possibilité d'une metadescription fonctionnelle et non fonctionnelle ;
- Le polymorphisme possible d'un interacteur ;
- Son éventuelle dotation de mécanismes d'adaptation.

Si cette approche règle le levier du remodelage, le problème de la redistribution reste ouvert. Notre hypothèse est celui d'une encapsulation, dans l'interface finale, d'interacteurs doués de capacités de redistribution. C'est le cas des IAMInteractor (Lachenal, 2004).

L'approche étant précisée, les hypothèses de travail étant énoncées, la section suivante traite de la conception des comets.

## 4 Les comets sous l'angle de la conception

De façon très modeste, sans prôner l'auto-adaptation des interacteurs, les comets (Calvary et al., 2004) se veulent rectifier les insuffisances des boîtes à outils : aucune boîte à outils n'explique les descriptions fonctionnelles et non fonctionnelles des interacteurs. Par exemple, la sémantique des boutons radio (sélectionner 1 option parmi N) et des cases à cocher (sélectionner M options parmi N) reste implicite. Le savoir-faire du concepteur et l'attention qu'il portera au critère de compatibilité sont déterminants. C'est ainsi, que dans Word (Figure 7), des cases à cocher sont utilisées à la place de boutons radio : des options étant deux à deux exclusives (par exemple, barré et barré double), les cases à cocher ne sont pas appropriées.

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

Figure 7. Cases à cocher inappropriées pour des options deux à deux incompatibles.

Dans le même esprit, les relations fonctionnelles (Coutaz et al., 1995) entre interacteurs (complémentarité du libellé et de la liste déroulante en Figure 2b ; équivalence fonctionnelle des versions présentées en figure 2) restent implicites. Il en est de même pour le coût des interacteurs : aucune boîte à outils ne rend observables les requis des interacteurs en termes de ressources d'interaction (surface d'affichage, dispositif de pointage), d'effort humain (longueur de trajectoire d'interaction), etc. Les boîtes à outils supposent toutes une connaissance de la part des concepteurs et la mise à disposition de ces ressources à l'exécution.

Les comets se veulent corriger ces insuffisances et apporter une solution à la construction plus spécifique de systèmes interactifs plastiques.

### 4.1 Définition

Une comet est un interacteur autodescriptif, capable de s'adapter ou d'être adapté au contexte d'usage, capable d'être recruté (versus éliminé) lorsqu'il est apte (versus inapte) à couvrir le nouveau contexte d'usage. Un interacteur est une entité logicielle et/ou matérielle rendant possible à l'utilisateur la réalisation d'une tâche. Son usage suppose la mise à disposition de ressources de calcul, de communication et d'entrée/sortie pour la réalisation des actions physiques composant la tâche. L'autodescription porte sur les propriétés fonctionnelles et non fonctionnelles de la comet.

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

La section suivante propose une classification des comets.

### 4.2 Classification

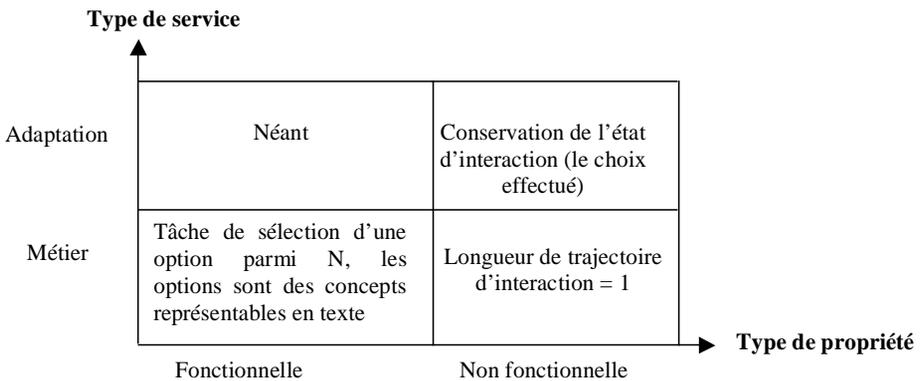
Nous distinguons trois types de comets : les comets autodescriptives, polymorphes et auto-adaptatives.

Dans leur forme la plus basique, les comets sont autodéscriptives, rendant observables leurs propriétés fonctionnelles et non fonctionnelles. L'autodescription peut s'appuyer sur des mécanismes d'introspection ou de réflexion (Englander, 2001). Les propriétés fonctionnelles définissent les services rendus par la comet, c'est-à-dire les tâches utilisateur qu'elle supporte et les concepts du domaine qu'elle manipule. Les propriétés non fonctionnelles qualifient le service rendu en en précisant, par exemple, la performance ou la tolérance aux fautes (Chung, 1991) (Rosa et al., 2002). L'autodescription (fonctionnelle et non fonctionnelle) est dynamique reflétant ainsi, par exemple, le niveau d'accomplissement d'une tâche.

Nous distinguons deux types de propriétés fonctionnelles selon qu'elles relèvent des capacités métier de la comet (par exemple, afficher l'heure) ou de ses capacités d'adaptation (par exemple, se remodeler au redimensionnement de la fenêtre). Dans leur forme la plus basique, les comets ne sont pas armées pour l'adaptation. Elles peuvent embarquer des mécanismes d'adaptation (reconnaissance du contexte d'usage comme dans Dey et al. (1999), calcul et/ou mise en œuvre de la réaction – Cf. section 3.1) mais elles se doivent alors de décrire cette capacité d'adaptation. Dans PlasticClock, les trois mécanismes sont implémentés pour une adaptation à la taille de la fenêtre.

Les propriétés non fonctionnelles peuvent, de la même façon, être déclinées selon qu'elles relèvent des capacités métier ou d'adaptation de la comet. Elles s'expriment dans un référentiel donné (Cf. section 3.1). Par exemple, PlasticClock pourrait être évalué en termes de sûreté de fonctionnement (métier) et de continuité de l'interaction (adaptation). Il est à noter que des requis non fonctionnels d'adaptation peuvent être formulés même si la comet n'est pas auto-adaptative. En effet, ces propriétés pourraient instruire une adaptation externe (Cf. section 3.1).

La figure 8 résume cette classification de l'autodescription. Elle l'illustre sur la comet bouton radio minimale, c'est-à-dire ni polymorphe, ni auto-adaptative.



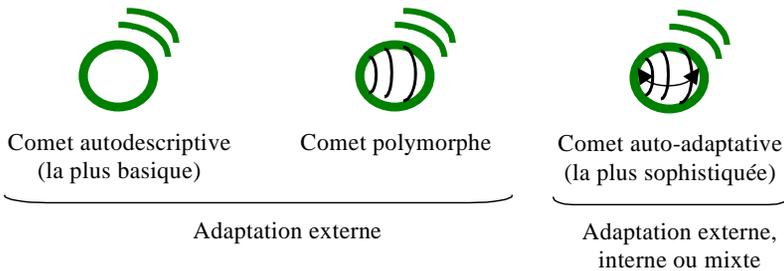
**Figure 8.** Classification des propriétés fonctionnelles et non fonctionnelles selon qu'elles relèvent des capacités métier ou d'adaptation de la comet. La classification est illustrée sur la comet bouton radio minimale (ni polymorphe, ni auto-adaptative).

Les comets polymorphes sont une version élaborée des comets : elles disposent d'un ensemble d'incarnations. Ce potentiel peut être précieux en cas de changement de contexte d'usage, les incarnations pouvant se distinguer dans les ressources qu'elles requièrent. Le polymorphisme peut s'exercer à différents niveaux d'abstraction : concepts et tâches, interfaces abstraite, concrète et finale (Cf. section

3.1). PlasticClock, par exemple, est polymorphe : elle embarque les présentations des figures 3 et 4. Le polymorphisme s’y exerce aux niveaux tâche (jouant sur la disponibilité de la tâche « consulter date » seulement offerte en version Figure 3d), interfaces abstraites (par le partage ou non d’un même espace de travail pour représenter les deux heures), concrètes (par le choix de rendu texte/cadran) et finales (par l’usage de boîtes à outils spécifiques pour l’exécution sur PC ou téléphone). En cas de changement de contexte d’usage, PlasticClock, dans son adaptation interne, joue sur la présentation qu’elle choisit d’afficher. Il est à noter que les travaux en multimodalité sont ici à considérer, les propriétés CARE (Complémentarité, Assignation, Redondance, Equivalence) (Coutaz et al, 1995) s’appliquant en particulier. On peut, par exemple, décider de recourir simultanément à une présentation graphique et vocale (redondance) pour pallier un environnement bruyant.

Les comets auto-adaptatives (adaptation interne) sont les comets les plus riches fonctionnellement. Elles embarquent tous les mécanismes nécessaires à leur adaptation (reconnaissance du contexte d’usage, calcul et mise en œuvre de la réaction – Cf. section 3.1). La réaction peut mettre à profit l’éventuel polymorphisme de la comet.

La figure 9 résume cette classification des comets. Elle précise pour chaque type (autodescriptif, polymorphe, auto-adaptatif), la nature de l’adaptation (interne, externe ou mixte).



**Figure 9.** Classification des comets et localisation de l’adaptation.

La section suivante propose une modélisation couvrant les trois gammes de comets (autodescriptives, polymorphes, auto-adaptatives).

#### 4.3 Modélisation

Conformément aux acquis en plasticité (section 3.1), une comet se définit à quatre niveaux d’abstraction (Figure 10) : concepts et tâches (dit « abstraction » de la comet), interfaces abstraite, concrète et finale. Les niveaux se correspondent par réification (inversement abstraction). L’abstraction exporte les concepts et tâches que la comet représente. L’interface abstraite publie sa structuration en espaces de travail. L’interface concrète exporte son style (par exemple, le style bouton) et précise si ce style est typique dans le contexte d’usage considéré. L’interface finale (i.e. la version exécutable ou interprétable de la comet) gère la capture/réception/diffusion du contexte d’usage ainsi que l’état d’interaction de la comet. Typiquement, en cas de redistribution, elle saura s’arrêter, sonder son état d’interaction, puis éventuellement redémarrer dans ce même état après migration.

La comet est définie par au moins une abstraction. Elle peut ne pas avoir été réifiée, c’est-à-dire ne disposer d’aucune interface abstraite, concrète ou finale préfabriquée. Chaque niveau est doué d’autodescription. Il décrit, en particulier, sa dépendance en termes de contexte d’usage, de modalité d’interaction, d’interacteur

et d'environnement d'exécution (Cf. section 3.1). Il décrit aussi sa qualité d'usage pour un contexte d'usage et une propriété donnée au regard d'un référentiel qualité.

La comet peut embarquer des modèles d'évolution et de transition pour régir son adaptation. La comet sait (et l'exporte) si elle est polymorphe sur un ensemble de contextes donnés, auto-adaptative et/ou plastique pour un ensemble de propriétés données.

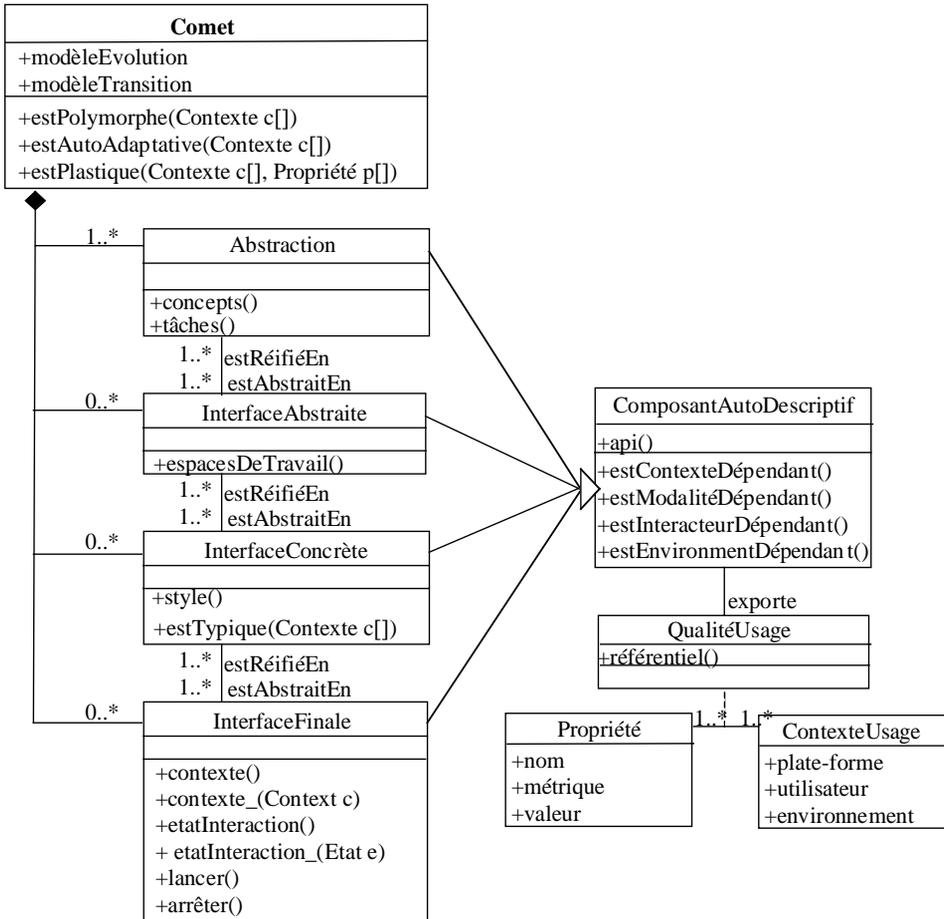


Figure 10. Modélisation des comets.

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

## 5 Les comets sous l'angle de l'exécution

Le système interactif est ici vu comme une composition de comets (autodescriptives, polymorphes et/ou auto-adaptatives). Cette section discute des stratégies et politiques d'adaptation pour le calcul de la réaction (lettre C de la figure 5). Le calcul peut être fait de façon statique ou dynamique. Il fait suite à un changement de contexte d'usage détecté par le C de la figure 5. Les stratégies et politiques sont ici étudiées sous l'angle de la comet.

## 5.1 Stratégies

Nous identifions quatre classes de stratégies :

- L'adaptation par polymorphisme. Cette stratégie n'a de sens que pour des comets polymorphes : la comet est adaptée au contexte d'usage par changement de sa *forme* (par exemple, la transformation d'un jeu de boutons radio en une liste déroulante). Cette stratégie conserve la comet (son instance) : elle en change l'incarnation. En termes de niveau d'abstraction, le changement peut s'opérer à des profondeurs variables : concepts et tâches, interfaces abstraite, concrète et/ou finale. L'adaptation peut s'effectuer selon trois cardinalités, 1-1, 1-N, N-1 selon qu'une incarnation est remplacée par une autre (cardinalité 1-1, par exemple, l'interface concrète Figure 4a est remplacée par l'interface concrète Figure 4b), N autres (cardinalité 1-N) ou que N incarnations sont agrégées en une seule (cardinalité N-1). Typiquement, dans la figure 2, la commutation des boutons radios en une liste déroulante est un exemple de polymorphisme 1-1 ; l'englobement du libellé dans la liste déroulante est un exemple de polymorphisme N-1 ; inversement, l'externalisation du libellé est un exemple de polymorphisme 1-N ;
- L'adaptation par substitution. Par opposition au polymorphisme, cette stratégie ne préserve pas la comet. La comet est détruite : elle est remplacée par une 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 sur la base de comets non polymorphes ;
- L'adaptation par ajout. Une comet est recrutée pour, par exemple, offrir une nouvelle tâche ou renforcer une information déjà présentée (représentation multiple d'un même concept). Si PlasticClock n'était pas polymorphe, l'ajout de la date (Figure 3d) nécessiterait le recrutement d'une comet « consulter date » ;
- L'adaptation par suppression : c'est la duale de l'ajout. Une comet est supprimée parce que la tâche ne fait plus sens ou ne peut plus être offerte dans le contexte d'usage courant.

A l'exécution, les stratégies sont dictées par le modèle d'évolution de la comet (ou des autres comets), sous le contrôle de la qualité d'usage requise. Les stratégies sont déployées selon des politiques présentées dans la section suivante.

## 5.2 Politiques

Les politiques dépendent de l'autonomie accordée à la comet dans sa prise en charge de l'adaptation : reconnaissance du contexte d'usage, calcul et mise en oeuvre de la réaction. Pour chacun de ces aspects, un degré d'autonomie est accordé à la comet, donnant lieu à trois classes de politiques :

- La politique *Non concertée externe* correspond à une autonomie nulle : la comet est incapable (pas d'adaptation interne) ou jugée inapte à gérer l'adaptation. L'adaptation est prise en charge par un tiers (adaptation externe assurée par une autre comet ou l'infrastructure d'exécution) ;
- La politique *Non concertée interne* correspond à une autonomie totale : la comet est douée d'adaptation et 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. Ceci peut être incompatible avec la qualité d'usage requise.

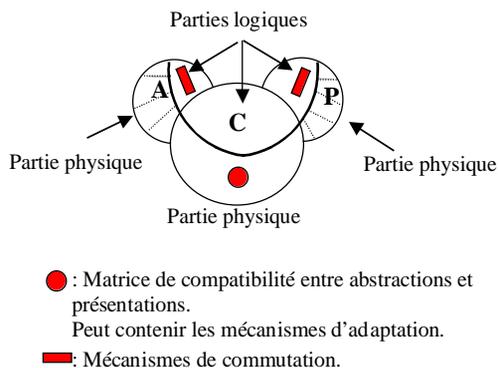
Le choix de la politique se fera au regard de la qualité d'usage requise. Seule la politique non concertée interne est aujourd'hui implémentée. Les autres font partie des perspectives.

## 6 Mise en œuvre

L'objectif visé est la mise à disposition d'un *atelier de plasturgie* pour la confection de comets et l'élaboration de systèmes interactifs plastiques. Aujourd'hui, l'effort de réalisation s'est porté sur l'architecture logicielle. Les comets sont implémentées selon le modèle Compact (Context Mouldable PAC for Plasticity) (Calvary et al., 2004). Compact est une déclinaison du modèle PAC (Coutaz, 1987) pour la plasticité. PAC est un modèle à agents reconnaissant trois facettes aux constituants des systèmes interactifs : une abstraction, une présentation et un contrôle. Le contrôle assure la cohérence et la communication entre les facettes abstraction et présentation. Pour une séparation des préoccupations, Compact « tranche » chaque facette en deux, isolant ainsi en chacune d'elles une partie logique et une partie physique (Figure 11) :

- Le logique gère l'état de la facette indépendamment de son incarnation. Il mémorise, par exemple, le choix de consultation des comptes bancaires que ce choix ait été fait par liste déroulante ou bouton radio. Il connaît l'ensemble des alternatives possibles (abstractions ou présentations – elles sont multiples en cas de polymorphisme) et embarque les mécanismes nécessaires à leur exploitation ;
- Le physique est l'incarnation courante de la facette.

Appliquée à la facette abstraction, cette séparation donne lieu à une abstraction logique et une abstraction physique. L'abstraction physique gère des données et traitements qui lui sont spécifiques, selon des formats et qualité qui lui sont propres (les algorithmes différeront typiquement en précision et temps de calcul). L'abstraction logique maintient l'état courant *pivot* de la comet et embarque les mécanismes nécessaires à la commutation entre abstractions physiques.



**Figure 11.** Le modèle d'architecture logicielle Compact.

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

Dans le même esprit, le contrôle logique gère la cohérence et la communication entre les abstraction et présentation logiques. Il est, en pratique, mis en œuvre par un contrôle physique. Le contrôle physique est en charge de (a)

recevoir et/ou capter et/ou diffuser le contexte d'usage, (b) recevoir et/ou calculer et/ou diffuser la réaction à mettre en œuvre, (c) éventuellement mettre en œuvre la réaction. La réaction peut consister en le changement d'abstraction et/ou de présentation physiques. Le calcul s'appuie sur une matrice pré-établie de compatibilité entre abstractions et présentations physiques.

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

Comme dans PAC, un système interactif devient une collection d'agents Compact. Des canaux de communication spécifiques peuvent être établis entre les contrôles physiques pour une propagation particulière du contexte d'usage : par exemple, une gestion centralisée de la cohérence ergonomique. Les contrôles logiques sont alors reliés à un même père gérant l'adaptation.

Compact est implémenté et aujourd'hui éprouvé sur un ensemble restreint de comets : les spécificateurs (1 parmi N), conteneurs (espaces de travail incarnés en fenêtres ou canevas) et navigateurs (nuls, boutons, onglets, liens). Dans nos perspectives à court terme, figure la confrontation de ce modèle avec des modèles existants tels qu'AMF (Tarpin-Bernard et David, 1999), par exemple, pour son aspect multi-facettes.

## **7 Conclusion et perspectives**

Cet article propose une nouvelle génération de « widgets », les comets, pour la construction de systèmes interactifs plastiques. Au delà de l'adaptation, c'est une amélioration des boîtes à outils actuelles qui est aussi visée. Il s'agit de rendre explicites les propriétés fonctionnelles et non fonctionnelles des interacteurs.

En premier lieu, les comets se distinguent des interacteurs par leur capacité d'autodescription. Elles peuvent s'enrichir de polymorphisme et s'êtouffer de capacités d'adaptation mais ceci n'est pas nécessaire pour être comet. L'autodescription en est le seul requis. Cette classification des comets résulte de l'approche adoptée.

L'approche marie deux grains d'analyse, les systèmes interactifs et les interacteurs. Faisant l'hypothèse d'une granularité variable des systèmes interactifs et des interacteurs, l'originalité est ici l'application, à petit grain (celui des interacteurs), des acquis en plasticité. En découlent la classification des comets et leur modélisation. L'approche semble raisonnable et puissante, l'état de l'art des interacteurs devenant des instanciations ou des couvertures particulières des comets. Une évaluation expérimentale est nécessaire pour juger de l'intérêt du polymorphisme et des capacités d'adaptation. C'est notre priorité. Une boîte à outils de comets est en cours de construction. Ces expériences permettront aussi d'évaluer la pertinence des stratégies et politiques d'adaptation. Une révision du modèle d'architecture logicielle pourra s'ensuivre condamnant, par exemple, le canal de communication spécifique à l'adaptation.

Les comets, par leur aspect multi-échelle, embrassent l'ingénierie des systèmes interactifs et en couvrent, en conséquence, les nombreux pans de recherche. Nos priorités à court terme portent sur :

- les propriétés non fonctionnelles : comment spécifier, mesurer et gérer la qualité d'usage ? ;
- l'architecture logicielle ;
- et le modèle d'évolution pour la conduite et l'apprentissage des réactions. Une modélisation multi-niveaux du système interactif sous forme de graphe est en cours de réalisation. L'adaptation deviendrait alors une transformation de ce

modèle, rejoignant ainsi (encore) les approches basées modèles. Une étroite collaboration est menée avec la communauté Ingénierie Dirigée par les Modèles (IDM, MDE, MDA).

A plus long terme, il s'agit bien sûr d'ouvrir la plasticité sur les interfaces multimodales dans la lignée de Crease (2001). La multimodalité serait alors vue comme un moyen d'adaptation, requérant la description fonctionnelle et non fonctionnelle de chaque modalité. De façon plus lointaine, on pourrait aussi envisager la dimension collective... Mais pour le moment, notre priorité porte résolument sur l'intégration de nos modèles de comet et d'évolution au langage UsiXML (<http://www.usixml.org>) dans un objectif de standardisation.

## 8 Remerciements

Ce travail a été réalisé dans le cadre du projet européen CAMELEON R&D IST2000-30104. Il se poursuit dans le NoE SIMILAR WP 8. Les auteurs remercient particulièrement Jean Vanderdonckt, membre des deux projets.

## 9 Références

Abowd, G.D., Coutaz, J., Nigay, L. (1992). Structuring the Space of Interactive System Properties. In *Engineering for Human-Computer Interaction*, J. Larson and C. Unger (Eds.), Elsevier Science Publishers B.V. (North-Holland), IFIP, 113-126.

Bastien, J.M.C., Scapin, D.L. (1993). *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*. Technical report, N° 156, INRIA Rocquencourt, Version V2.1, May 1993.

Calvary, G., Coutaz, J., Thevenin, D. (2001). Supporting Context Changes for Plastic User Interfaces : a Process and a Mechanism. In *People and Computers XV – Interaction without Frontiers, Joint Proceedings of AFIHM-BCS Conference on Human-Computer Interaction IHM-HCI'2001 (Lille, 10-14 September 2001)*, A. Blandford, J. Vanderdonckt and Ph. Gray (Eds.), Vol. I, Springer-Verlag, London, 349-363.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. *Interacting With Computers*, Vol. 15/3, 289-308.

Calvary, G., Coutaz, J., Dâassi, O., Balme, L., Demeure, A. (2004). Towards a new generation of widgets for supporting software plasticity: the « comet ». In *Proceedings of EHCI-DSVIS'2004, The 9th IFIP Working Conference on Engineering for Human-Computer Interaction Jointly with the 11th International Workshop on Design, Specification and Verification of Interactive Systems*, Hamburg, Germany, July 11-13.

Chung, L. (1991). Representation and Utilization of Non-functional Requirements for Information System Design. In *Proceedings of the 3rd international conference on Advanced Information System Engineering – CaiSE'91*, Trondheim, Norway, May 1991.

Constantine, L.L., Lockwood, L.A.D. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centred Design*. New-York, Addison-Wesley.

Coutaz, J. (1987). PAC, an Object Oriented Model for Dialog Design. In *INTERACT 87 - 2nd IFIP International Conference on Human-Computer Interaction (September 1-4, 1987, Stuttgart, Germany)*, H.J. Bullinger, B. Shackel (Eds.), 431-436, North-Holland.

- Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., Young, R. (1995). Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE properties. In *Proceedings of the INTERACT'95 conference (Lillehammer, Norway, June 25-29, 1995)*, S.A. Arnesen and D. Gilmore (Eds.), Chapman & Hall Publ., pp. 115-120.
- Crease, M. (2001). *A Toolkit Of Resource-Sensitive, Multimodal Widgets*. PhD Thesis, Department of Computing Science, University of Glasgow, December.
- Dâassi, O., Calvary, G., Coutaz, J., Demeure, A. (2003). Comet : Une nouvelle génération de « Widget » pour la Plasticité des Interfaces. In *Actes de la 15ème conférence francophone IHM 2003 (Caen, 24-28 Novembre 2003)*, ACM International Conference Proceedings Series, 64-71, ACM Press.
- De Abreu-Cybis, W. (1994). Vers un modèle d'objet d'interaction abstrait pour le raisonnement ergonomique sur l'interface homme-machine. In *Actes du XXIX Congrès de la Société d'Ergonomie de Langue Française (Paris, Maison de la Chimie, 21-23 septembre 1994)*, Coll. de la Direction des Etudes et Recherches d'Electricité de France, Vol. 2 Ergonomie & Ingénierie, Eyrolles, Paris, 162-169.
- Demeure, A., Calvary, G. (2003). Le Modèle d'Evolution en Plasticité des Interfaces : Apport des Graphes Conceptuels. In *Actes de la 15ème conférence francophone IHM 2003 (Caen, 24-28 Novembre 2003)*, ACM International Conference Proceedings Series, 80-87, ACM Press.
- De Palma, N., Bellisard, L., Riveill, M. (1999). Dynamic Reconfiguration of Agent-Based Applications. In *Proceedings Third European Research Seminar on Advances in Distributed Systems (ERSADS'99)*, Madeira Island, Portugal.
- Dey, A.K., Salber, D., Futakawa, M., Abowd, G.D. (1999). *An Architecture To Support Context-Aware Applications*. GVU Technical Report GIT-GVU-99-23, June.
- Dix, A., Finlay, J., Abowd, G., Beale, R. (1993). *Human-Computer Interaction*. Prentice-Hall, New-Jersey, 1993.
- Duke, D.J., Harrison, M.D. (1993). Abstract Interaction Objects. In *Proceedings of Eurographics'93*, R.J. Hubbard and R. Juan (Eds.), Computer Graphics Forum, Vol. 12, No. 3, Eurographics Association, Blackwell Publishers, 1993, 25-36.
- Englander, R. (2001). *Developing Java Beans*. O'Reilly publ., 316 pages.
- Faconti, G., Paterno, F. (1990). An approach to the formal specification of the components of an interaction. In *Proceedings Eurographics'90*, C. Vandoni and D. Duce (Eds.), 481-494, North-Holland.
- Farenc, C., Palanque, P., Vanderdonckt, J. (1995). User Interface Evaluation: is it Ever Usable?. In *Symbiosis of Human and Artifact: Human and Social Aspects of Human-Computer Interaction, Proceedings of 6th International Conference on Human-Computer Interaction HCI International'95 (Yokohama, Japon, July 9-14, 1995)*, Y. Anzai, K. Ogawa and H. Mori (Eds.), Advances in Human Factors/Ergonomics Series, Vol. 20B , Elsevier Science B.V., Amsterdam, 1995, 329-334.
- Garlan, D., Schmerl, B., Chang, J. (2001). Using Gauges for Architectural-Based Monitoring and Adaptation. In *Working Conference on Complex and Dynamic Systems Architecture*, Brisbane, Australia, December.
- Grolaux, D., Van Roy, P., Vanderdonckt, J. (2001). QtK: An Integrated Model-Based Approach to Designing Executable User Interfaces. In *PreProc. of 8th Int.*

*Workshop on Design, Specification, Verification of Interactive Systems DSV-IS'2001 (Glasgow, June 13-15, 2001)*, Ch. Johnson (Ed.), GIST Tech. Report G-2001-1, Dept. of Comp. Sci., Univ. of Glasgow, Scotland, 77-91. Accessible at [http://www.dcs.gla.ac.uk/~johnson/papers/dsvvis\\_2001/grolaux](http://www.dcs.gla.ac.uk/~johnson/papers/dsvvis_2001/grolaux).

Grundy, J, Yang, B. (2003). An environment for developing adaptive, multi-device user interfaces. In *Fourth Australia User Interface Conference (AUI2003)*, Adelaide, Australia, Conferences and practice in information Technology, Vol. 18, Robert Biddle and Bruce Thomas, Eds. Reproduction for academic.

IFIP (1996). *Design Principles for Interactive Software*. IFIP WG 2.7 (13.4), C. Gram and G. Cockton (Eds.), Chapman & Hall Publ.

ISO/IEC CD 25000.2 (2003a). *Software and Systems Engineering – Software product quality requirements and evaluation (SquaRE) – Guide to SquaRE*, 2003-01-13.

ISO/IEC 25021 (2003b). *Software and System Engineering – Software Product Quality Requirements and Evaluation (SquaRE) – Measurement*, 2003-02-03.

Lachenal, C. (2004). *Modèle et Outil pour l'Interaction multi-instrument et multisurface*. Thèse de l'Université Joseph-Fourier, Grenoble I, Spécialité Informatique.

Lopez-Jaquero, V., Montero, F., Molina, J.P., Gonzalez, P. (2004). A Seamless Development Process of Adaptive User Interfaces Explicitly Based on Usability Properties. In *Proceedings of EHCI-DSVIS'2004, The 9th IFIP Working Conference on Engineering for Human-Computer Interaction Jointly with the 11th International Workshop on Design, Specification and Verification of Interactive Systems*, Hamburg, Germany, July 11-13.

Markopoulos, P. (2001). Interactors: formal architectural models of user interface software. In *Encyclopedia of Microcomputers, Volume 27 (Supplement 6)*, A. Kent and J.G. Williams (Eds.), Marcel Dekker, New York, 203-235.

Montero, F., Vanderdonckt, J., Lozano M. (2004). Quality Models for Automated Evaluation of Web Sites Usability and Accessibility. In *Proceedings of the International Conference on Web Engineering, ICWE'2004*, July 28-30, Munich.

Mori, G., Paternò, F., Santoro, C. (2003). Tool Support for Designing Nomadic Applications. In *Proceedings of IUI 2003*, Miami, Florida, January 12-15.

Morin, D. (1990). Working Group Discussion: Current Practice. In *User Interface Management and Design, Proceedings of the Workshop on User Interface Management Systems and Environment (Lisbonne, 4-6 Juin 1990)*, D.A. Duce, M.R. Gomes, F.R.A. Hopgood and J.R. Lee (Eds.), Eurographics Seminars Series, Springer-Verlag, Berlin, 51-56.

Nielsen, J. (1994). Heuristic evaluation. In *Usability Inspection Methods*, J. Nielsen and R.L. Mack (Eds.), John Wiley & Sons, New York, NY.

Oreizy, P., Gorlick, M.M., Taylor, R.N., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D.S., Wolf, A. (1999). An Architecture-Based Approach to Self-Adaptive Software. *IEEE Intelligent Systems*, vol. 14, N. 3, 54-62.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T. (1994). *Human-Computer Interaction*. Wokingham, UK, Addison Wesley Publishing Company.

Rekimoto, J. (1997). Pick and Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST97*, ACM Press, 31-39.

- Rosa, N.S., Cunha, P.R.F., Justo, G.R.R. (2002). ProcessNFL: A Language for Describing Non-Functional Properties. In *Proceedings of the 35<sup>th</sup> Hawaii International Conference on System Sciences*, IEEE Press, 2002.
- Schmidt, A. (1999). Implicit human-computer interaction through context. In *Proceedings 2th Workshop on Human Computer Interaction with Mobile Devices*, Edinburgh, Scotland, 31 August 1999.
- Shneiderman, B. (1997). *Designing User Interface Strategies for effective Human-Computer Interaction* (3rd ed). Addison-Wesley Publishing Company.
- Seffah, A., Kececi, N., Donyaee, M. (2001). QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models. In *Proceedings APAQS Second Asia-Pacific Conference on Quality Software*, December, Hong-Kong, 10-11.
- Shackel, B. (1991). Usability-Context, framework, definition, design and evaluation. In *Human factors for informatics usability*, B. Shackel and S. J. Richardson (Eds.), Cambridge: Cambridge University, 21-37.
- Stephanidis, C., Akoumianakis, D. (2001). Universal design: Towards universal access in the Information society. In *Proceedings CHI 2001*, 31 March- 5 April. Workshop.
- Tarpin-Bernard, F., David, B.T. (1999). AMF : un modèle d'architecture multi-agents multi-facettes. *Techniques et Sciences Informatiques*, vol. 18, n. 5, 555-586.
- Thevenin, D., Coutaz, J. (1999). Plasticity of User Interfaces: Framework and Research Agenda. In *Proceedings Interact'99, Edinburgh*, A. Sasse and C. Johnson (Eds.), IFIP IOS Press, 110-117.
- Thevenin, D. (2001). *Adaptation en Interaction Homme-Machine : Cas de la plasticité*. Thèse de doctorat de l'Université Joseph Fourier, Décembre.
- Vanderdonckt, J., Bodart, F. (1993). Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. In *Proceedings of the Conference on Human Factors in Computing Systems InterCHI,93 « Bridges Between Worlds », Amsterdam (Pays-Bas), 24-29 avril 1993*, S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), ACM Press, New York, 424-429.
- Vanderdonckt, J. (1997). *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.
- Van Welie, M., van der Veer, G.C., Eliëns, A. (1999). Usability Properties in Dialog Models: In *Proceedings 6th International Eurographics Workshop on Design Specification and Verification of Interactive Systems DSV-IS'99*, Braga, Portugal, 2-4 June 1999, 238-253.