

Métamodèle de Règles d'Adaptation pour la Plasticité des Interfaces Homme-Machine

Vincent Ganneau^{1,2}

Gaëlle Calvary²

Rachel Demumieux¹

¹ France Télécom Division R&D
2, avenue Pierre Marzin
22307, Lannion Cedex, France
{vincent.ganneau, rachel.demumieux}
@orange-ftgroup.com

² Laboratoire LIG
385, rue de la Bibliothèque, BP 53
38041, Grenoble Cedex, France
{vincent.ganneau, gaelle.calvary}
@imag.fr

RESUME

En informatique ambiante, l'ordinateur conventionnel disparaît au profit d'un espace interactif au sein duquel les ressources d'interaction (PC, PDA, téléphone, Smartphone, etc.) sont variées et dynamiques. Une Interface Homme-Machine (IHM) est dite plastique si elle est capable de s'adapter à son contexte d'usage (utilisateur, plate-forme, environnement) dans le respect de sa valeur. L'article traite de la plasticité des IHM. Il présente une décomposition fonctionnelle de l'adaptation puis en développe trois aspects : le contexte d'usage, le moteur d'adaptation et le modèle utilisateur. Dans une approche IDM (Ingénierie Dirigée par les Modèles), un métamodèle de règles d'adaptation est proposé. Ce métamodèle aide non seulement les concepteurs à se poser les bonnes questions mais soutient aussi une instrumentation automatique du code pour l'observation du contexte d'usage, l'application des réactions et l'éventuelle révision, par apprentissage, des règles d'adaptation.

MOTS CLES : Plasticité des Interfaces Homme-Machine, adaptation, contexte d'usage, valeur, règle d'adaptation, métamodèle, modèle, transformation, IDM.

ABSTRACT

In ubiquitous computing, platforms are no more limited to conventional computers: users interact with interactive spaces made of various and dynamic interaction resources (PC, PDA, phone, smartphone, etc.). In Human-Computer Interaction, plasticity denotes the capacity of User Interfaces (UIs) to withstand variations of context of use (user, platform, environment) while preserving their value. This paper presents a functional decomposition of plasticity. It focuses on the context of use, adaptation engine and user modeling. According to a Model Driven Engineering (MDE) approach, it proposes a

metamodel of adaptation rules for both helping the designers in reasoning on plasticity, and sustaining an automatic instrumentation of the UI for probing the context of use, applying the reaction, and possibly revising the adaptation rules through learning.

CATEGORIES AND SUBJECT DESCRIPTORS: D.2.2 [Software Engineering]: Design Tools and Techniques|User Interfaces; H.5.2 [Information Interfaces and Presentation]: User Interfaces|User-centered design.

GENERAL TERMS: Design, Human Factors

KEYWORDS: Plasticity of User Interfaces, adaptation, context of use, value, adaptation rule, metamodel, model, transformation, MDE.

INTRODUCTION

Avec les avancées des réseaux sans fil et les progrès en miniaturisation, l'informatique ambiante [31] devient réalité : l'utilisateur est mobile, il évolue dans un environnement varié et recourt, de façon opportuniste, à des plates-formes d'interaction diverses (PC, PDA, téléphone, etc.). Si cette vision séduit du point de vue de l'usage, elle pose des défis en ingénierie de l'interaction homme-machine : le contexte d'usage (utilisateur, plate-forme, environnement) n'est plus fixe, connu à la conception, mais devient varié, variable et potentiellement imprévisible. Ce constat a motivé en 1998 l'introduction d'une nouvelle propriété : la plasticité des Interfaces Homme-Machine (IHM) [29]. La définition s'est peu à peu affinée, désignant désormais la capacité d'une IHM à s'adapter à son contexte d'usage dans le respect de sa valeur [14]. La valeur peut être fonctionnelle (préservé un service) – elle relève alors de l'utilité du système interactif – et/ou non fonctionnelle (minimiser les actions physiques) – elle relève alors de l'utilisabilité.

La plasticité est aujourd'hui traitée au plan international. On distingue deux courants : les approches ascendantes qui s'ingénient à plastifier des applications existantes [27] [30] ; les descendantes qui prévoient la plasticité à la conception [5] [18]. Ces approches sont complémentaires et pourraient se rejoindre en la notion de *modèle*.

De nombreux travaux portent sur la description d'une IHM à différents niveaux d'abstraction, pour exemple UsiXML. Si ces travaux ont été poussés par la diversité des contextes d'usage, paradoxalement les règles d'adaptation ont été moins traitées. Cet article propose un métamodèle de règles d'adaptation à visée descriptive et générative [4]. Il s'agit, d'un point de vue conceptuel, d'aider le concepteur à se poser les bonnes questions en matière de plasticité puis de l'accompagner, d'un point de vue implémentatif, dans la mise en œuvre de l'adaptation.

La proposition est un outil *partenaire* instrumentant le code du système interactif conformément aux règles d'adaptation pour, non seulement, mettre en œuvre l'adaptation mais aussi en apprendre la pertinence et, si nécessaire, en réviser le modèle. Cette vision est en rupture avec les travaux jusqu'ici menés dans les hypermédias adaptatifs [32] et la génération d'IHM [23] pour lesquels les règles de transformation ont toujours été spécifiées en dur, compromettant, en conséquence, l'outil dès lors que ces règles sont jugées inconvenantes par l'utilisateur final.

L'article est structuré en quatre parties. La première propose un cas d'étude simple, à visée d'illustration. La seconde présente une décomposition fonctionnelle de l'adaptation. Trois fonctions sont examinées : la perception du contexte d'usage, la réaction et l'apprentissage. Elles font l'objet de sections. L'article se clôture sur un ensemble de perspectives.

CAS D'ETUDE : PLASTICSMS

Le succès du téléphone mobile n'est plus à démontrer. Pourtant, des évaluations auprès d'utilisateurs ont mis en évidence un certain nombre de difficultés d'utilisation [15] qui s'expliquent en partie par les problèmes de navigation et de saisie dus aux contraintes de ces terminaux (taille réduite de l'écran, clavier restreint). Ces problèmes s'accroissent avec la course à la couverture fonctionnelle croissante (visiophonie, télévision, photo, vidéo, etc.). Ils peuvent être des freins à l'usage. Malgré des innovations visant à simplifier les interactions utilisateur, comme par exemple la saisie prédictive [28] ou l'autocomplétion [25], les tâches de saisie restent longues et décourageantes.

PlasticSMS explore la voie de la plasticité pour la réception et l'envoi de messages textes (SMS). Il profite des ressources d'interaction environnantes (surface d'affichage, clavier, etc.) pour déconfiner l'IHM. La *redistribution* est placée sous le contrôle de l'utilisateur via une Méta-IHM [12], invocable par l'utilisateur (Figure 1a) ou affichée par le système selon sa perception du contexte d'usage (Figure 1b). L'IHM s'adapte aux plateformes qui l'hébergent : c'est la notion de *remodelage* (Figure 2). Sur PC, de nouvelles fonctionnalités ap-

paraissent, telles que des émoticônes, ou encore le choix d'une taille de police adaptée aux capacités visuelles de l'utilisateur. Le choix du destinataire y est facilité par un menu déroulant répertoriant les contacts enregistrés sur le téléphone. Ce menu déroulant est adaptatif : il réordonne les contacts en fonction de l'usage, plaçant en tête les contacts les plus utilisés.



Figure 1 : Méta-IHM pour contrôler la redistribution de l'IHM.

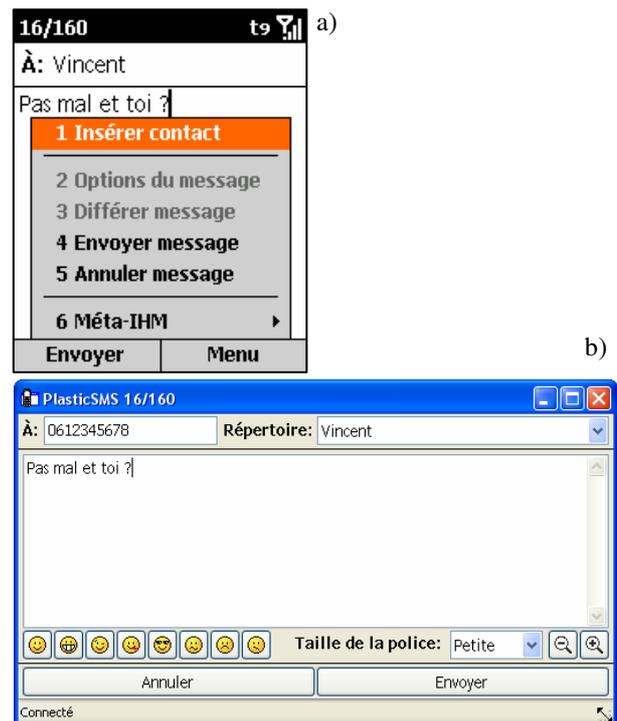


Figure 2: Remodelage de l'IHM selon la plate-forme cible (Smartphone en a ; PC en b).

Le prototype actuel tourne sur des Smartphones Orange™ équipés de Windows Mobile, système d'exploitation « ouvert » permettant le développement d'applications externes. La version actuelle explore uniquement la modalité graphique en sortie et la saisie clavier en entrée. La réception d'un message se fait sur le

mobile. La lecture de ce message peut être déportée sur toute autre plate-forme équipée d'un dispositif de sortie (son, écran, ...) et capable d'être notifiée par ce même téléphone. De même, la rédaction d'un SMS peut être déportée. En revanche, l'envoi des messages sur les réseaux GSM se fait uniquement depuis le téléphone mobile. Le prototype est en cours d'évaluation.

Remodelage et redistribution sont les deux leviers de la plasticité [9]. Les premiers travaux en plasticité (notamment le projet européen CAMELEON [2]) se sont focalisés sur la compréhension des leviers de l'adaptation. [26] montre que l'Ingénierie Dirigée par les Modèles (IDM) est une approche unificatrice permettant de couvrir remodelage et redistribution. Jusqu'ici l'effort a été porté sur la (méta)modélisation des IHM. Nous nous intéressons ici à la transformation de ces IHM quand le contexte d'usage change. La section suivante donne une décomposition fonctionnelle du processus d'adaptation.

DECOMPOSITION FONCTIONNELLE DE L'ADAPTATION

L'adaptation est vue comme un processus en trois étapes comprenant la reconnaissance du contexte d'usage, le calcul puis la mise en œuvre de la réaction [8]. La décomposition fonctionnelle de la Figure 3 intègre ces préoccupations. Elle précise que la réaction peut dépendre, bien sûr du contexte d'usage, mais aussi du système interactif ainsi que des préférences utilisateur explicites ou apprises. L'adaptation peut être placée sous le contrôle de l'utilisateur via une IHM dédiée : la méta-IHM.

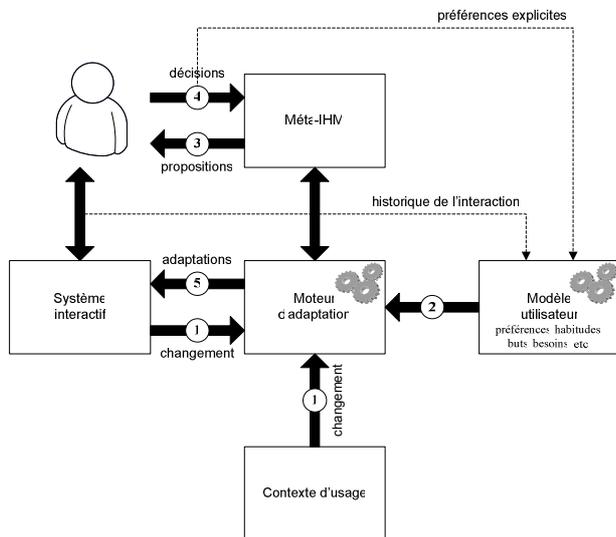


Figure 3 : Décomposition fonctionnelle de la plasticité.

L'article examine trois fonctions : la reconnaissance du contexte d'usage, l'adaptation proprement dite ainsi que la modélisation de l'utilisateur. Elles font l'objet des sections suivantes.

CONTEXTE D'USAGE

Nous reprenons l'ontologie de [11] pour la modélisation du contexte d'usage. Un contexte est défini par un ensemble d'entités (par exemple, le mobile), un ensemble de rôles que les entités peuvent tenir (par exemple, surface d'affichage), et un ensemble de relations entre entités (par exemple, est près de). Un changement de contexte survient lorsque l'un des ensembles (entités, rôles, relations) change. L'ensemble des contextes d'usage couverts par un système interactif est modélisé par un graphe dont les nœuds représentent les contextes d'usage et les arcs les conditions pour passer d'un contexte à l'autre.

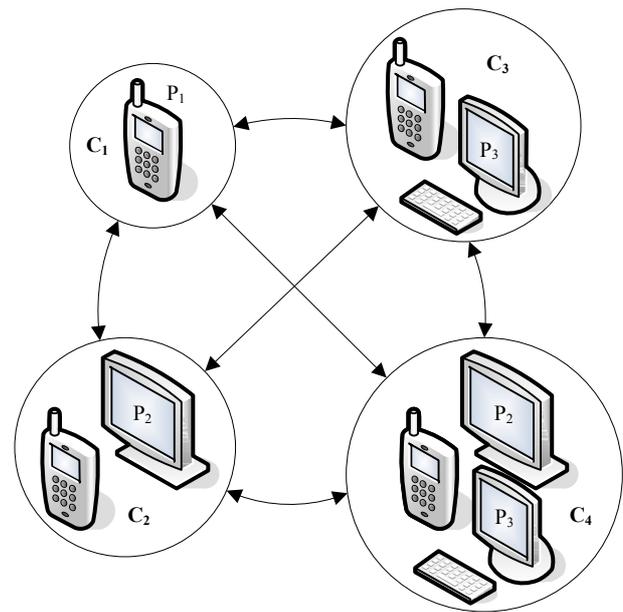


Figure 4 : Contextes et changements de contexte dans PlasticSMS. Les nœuds représentent les contextes d'usage. Les flèches dénotent les transitions entre contextes.

Dans PlasticSMS, on identifie quatre contextes d'usage (Figure 4) selon que l'interaction prend place dans la sphère domestique de l'utilisateur (cas des contextes C2, C3 et C4) ou lorsque celui-ci est en situation de mobilité (contexte C1). Pour chacun de ces contextes, l'ensemble des entités est constitué des plates-formes disponibles pour l'utilisateur. Précisons que le téléphone mobile est la plate-forme minimale requise pour faire fonctionner PlasticSMS. Les plates-formes peuvent jouer le rôle d'*instrument* (I) ou de *surface* (S) [22] selon qu'elles disposent ou non d'un dispositif d'entrée ou de sortie. Dans la suite, on note P1 le téléphone mobile assurant les rôles de surface et d'instrument ; P2 les plates-formes grand écran capables d'assurer le rôle de surface, typiquement un écran de téléviseur ; et P3 les plates-formes grand écran de type ordinateur jouant les rôles de surface et d'instrument, classiquement par la mise à disposition d'un clavier. L'ensemble des relations entre entités est constitué de relations de proximité entre les plates-

formes. Ce choix est motivé par la technologie Bluetooth® [6] utilisée dans PlasticSMS pour détecter et faire communiquer les différentes plates-formes en présence. Le graphe de contextes est un graphe complet : il est possible de passer d'un contexte à tout autre selon l'arrivée et le départ de plates-formes.

Nous proposons d'étendre l'ontologie de [11] en définissant des relations ensemblistes entre contextes d'usage. En mathématiques, la théorie des ensembles propose plusieurs opérations dont l'inclusion, l'union, l'intersection et la différence. Nous donnons ci-dessous les relations ensemblistes pour les contextes d'usage de PlasticSMS :

$$C_1 \subset C_2, C_1 \subset C_3, C_1 \subset C_4, C_2 \subset C_4, C_3 \subset C_4$$

$$(1) C_1 = C_2 \cap C_3, (2) C_4 = C_2 \cup C_3$$

Par définition, à chaque contexte C_i est associé un triplet (E_i, Ro_i, Rel_i) où E_i , Ro_i et Rel_i dénotent respectivement les ensembles d'entités, de rôles joués par les entités, et de relations entre entités impliqués dans la définition de ce contexte. A l'aide des relations ensemblistes (1) et (2), nous précisons chacun de ces ensembles pour les quatre contextes identifiés :

$$E_1 = \{P_1\}, E_2 = \{P_1, P_2\}, E_3 = \{P_1, P_3\}, E_4 = \{P_1, P_2, P_3\}$$

$$Ro_2 = \{(P_1, S), (P_1, I), (P_2, S)\},$$

$$Ro_3 = \{(P_1, S), (P_1, I), (P_3, S), (P_3, I)\}$$

$$Ro_1 = Ro_2 \cap Ro_3 = \{(P_1, S), (P_1, I)\}$$

$$Ro_4 = Ro_2 \cup Ro_3 = \{(P_1, S), (P_1, I), (P_2, S), (P_3, S), (P_3, I)\}$$

$$Rel_2 = \{estProche(P_1, P_2)\}$$

$$Rel_3 = \{estProche(P_1, P_3)\}$$

$$Rel_1 = Rel_2 \cap Rel_3 = \emptyset$$

$$Rel_4 = Rel_2 \cup Rel_3 = \{estProche(P_1, P_2), estProche(P_1, P_3)\}$$

L'opération de différence permet de déterminer la condition de changement entre deux contextes. Autrement dit, elle permet d'identifier les entités apparues ou disparues, les rôles nouvellement joués ou qui ne le sont plus, et les changements survenus dans les relations entre entités. Soient C_i et C_j des contextes tels que $C_i = (E_i, Ro_i, Rel_i)$ et $C_j = (E_j, Ro_j, Rel_j)$. Lors du passage de C_i à C_j , les ensembles d'entités apparues, de rôles nouvellement joués et de relations nouvellement formées sont donnés par :

$$C_j - C_i = (E_j - E_i, Ro_j - Ro_i, Rel_j - Rel_i)$$

Inversement, les ensembles d'entités disparues, de rôles n'étant plus joués et de relations rompues sont donnés par :

$$C_i - C_j = (E_i - E_j, Ro_i - Ro_j, Rel_i - Rel_j)$$

Prenons, par exemple, le passage de C_1 à C_2 dans PlasticSMS. On a :

$$E_2 - E_1 = \{P_2\}$$

$$Ro_2 - Ro_1 = \{(P_2, S)\}$$

$$Rel_2 - Rel_1 = \{estProche(P_1, P_2)\}$$

ce qui correspond bien à l'arrivée d'une seconde plate-forme jouant le rôle de surface d'affichage.

Les contextes d'usage étant identifiés, il s'agit à présent de spécifier les réactions à mettre en oeuvre.

MOTEUR D'ADAPTATION

Selon une approche IDM, nous plaçons au cœur de l'adaptation un métamodèle de règles d'adaptation. Ce métamodèle (Figure 5) s'appuie sur le paradigme Événement – Condition – Action (ECA) inspiré des bases de données actives [16] : lorsque l'événement survient, la condition est évaluée et, si elle est vérifiée, l'action est déclenchée :

on event if condition do action

L'événement déclencheur peut être un changement de contexte d'usage ou un événement système (redimensionnement d'une fenêtre, réception d'un nouveau message, etc.). La condition peut porter sur tous les observables : état du système interactif, contexte d'usage, historique de l'interaction personne-système, etc. L'action peut consister en une adaptation (remodelage ou redistribution) ou peut déclencher une autre règle d'adaptation. Une action peut satisfaire (ou ne pas satisfaire) un ensemble de propriétés. Par exemple, la redistribution d'une tâche sur plusieurs plates-formes pourra satisfaire la propriété de Redondance telle que définie dans les propriétés CARE [10]. Il suffira pour ce faire de rendre la tâche sur au moins deux plates-formes tenant le rôle de surface d'affichage et/ou d'instrument. On peut choisir d'autres cadres de référence pour expliciter des propriétés, par exemple le référentiel de Bastien et Scapin [3] pour l'utilisabilité.

Les règles d'adaptation sont conformes à leur métamodèle. Les syntaxes abstraite (M2A) et concrète (M2C) sont données en Figure 5. Par exemple, on peut écrire la règle AR00 suivante pour stipuler la migration de la tâche T2 (écrire un SMS) sur la plate-forme P3 (un PC) dès lors que l'utilisateur commence à écrire un SMS (sa tâche en cours Tc est T2) :

AR00: if Tc=T2 do T2/P3

Les règles de dégradation élégante introduites par [17] sont un cas particulier de règles d'adaptation. Elles traitent de la transformation d'IHM graphiques, conçues pour plates-formes luxueuses, en des IHM compatibles de ressources moindres. Dans l'approche, ces règles sont appliquées à la conception mais on pourrait transposer le travail à l'exécution.

Nous distinguons deux types de règles d'adaptation selon qu'elles répondent à un changement de contexte ou qu'elles adaptent le système interactif dans un contexte

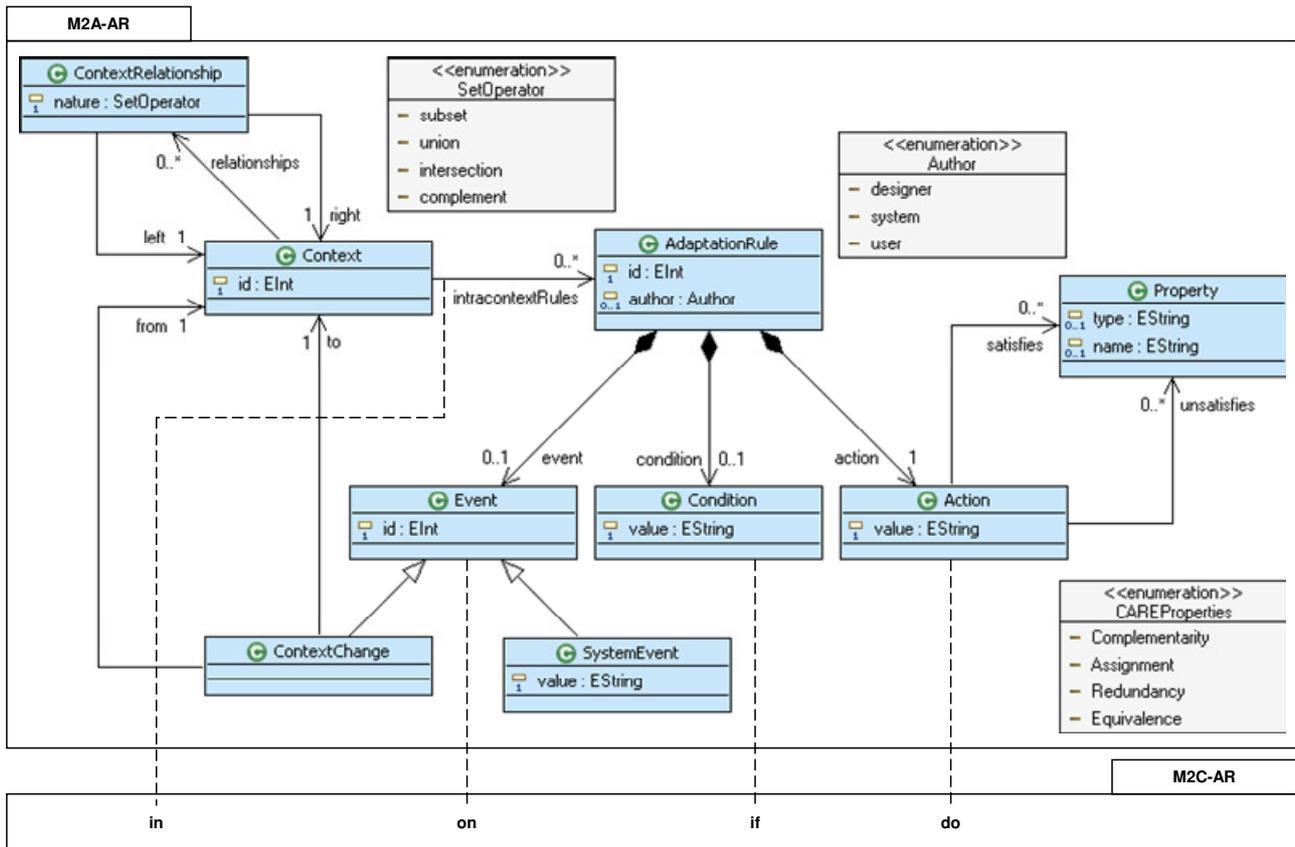


Figure 5 : Métamodèle de règles d'adaptation.

donné. Cette distinction donne lieu à des règles d'adaptation inter versus intra-contextes.

Règles d'adaptation inter-contextes. L'événement qui déclenche la règle est ici un changement de contexte. On peut ainsi préciser le modèle d'une règle d'adaptation inter-contextes :

```
on Ci->Cj if condition do action
```

Dans PlasticSMS, les règles d'adaptation inter-contextes sont les suivantes :

- AR01: on C1->C3 do AR00
- AR02: on C1->C4 do AR00
- AR03: on C2->C3 do AR00
- AR04: on C2->C4 do AR00
- AR05: on Ci->Cj if Tx/Py and Py≠Ej do ¬(Tx/Py)

Les règles AR01 à AR04 proposent de déporter la saisie du message sur une plate-forme moins contrainte. La partie action de ces quatre règles déclenche la règle AR00. AR05 est une règle générale dont l'objectif est de maintenir les mappings tâche-plate-forme dans un état cohérent en cas de changement de contexte d'usage : si une plate-forme Py disparaît alors qu'elle hébergeait la tâche Tx, alors il convient de cesser ce déploiement.

Règles d'adaptation intra-contextes. L'entête d'une règle d'adaptation intra-contexte définit le contexte d'usage dans lequel cette règle est active :

```
in Ci on event if condition do action
```

Dans PlasticSMS, les règles d'adaptation intra-contexte sont :

- AR06: in C2 on nouveau message do T1/P2
- AR07: in C3 on nouveau message do T1/P3
- AR08: in C3 on Tc->T2 do T2/P3
- AR09: in C4 on nouveau message do T1/P2 and T1/P3
- AR10: in C4 on Tc->T2 do T2/P3
- AR11: in Ci on redimension(lxh) do best(IHM(lxh))

Si l'on considère les relations ensemblistes définies précédemment, on peut remarquer que la partie action de la règle AR09 est l'union des parties action des règles AR06 et AR07, ces dernières ayant pour dénominateur commun la partie événement. Ici, la portée des règles AR06 et AR07 peut nous permettre de générer automatiquement la règle AR09 si l'on a défini précédemment les relations $C_2 \subset C_4$, $C_3 \subset C_4$ et $C_4 = C_2 \cup C_3$. De même, AR10 est déductible de AR08.

Vers une génération automatique du code. [4] évalue un modèle par rapport à son pouvoir descriptif et généra-

tif. Notre métamodèle (Figure 5) ne se veut pas seulement descriptif. Il vise aussi la génération du code de l'adaptation. Pour cela, nous définissons nos règles d'adaptation à l'aide d'un éditeur EMF (Eclipse Modeling Framework) qui produit en sortie le code XML correspondant. Ce code peut alors être manipulé par des outils de transformation de modèles tels qu'ATL [1].

```
<rule id="8">
  <event xsi:type="SystemEvent" id="6"
  value="nouveau message"/>
  <action value="T1/P3"/>
</rule>
<context id="3"
intracontextRules="//@rule.7 //@rule.8"/>
```

De cette spécification est généré l'extrait de code C# ci-dessous :

```
MessageInterceptor interceptor;
interceptor = new MessageInterceptor();
interceptor.MessageReceived += new
MessageInterceptorEventHandler(interceptor
_MessageReceived);
void interceptor_MessageReceived(object
sender, MessageInterceptorEventArgs e)
{
  message = (SmsMessage)e.Message;
  switch (context)
  {
    case 3: distribute(task1, platform3);
  }
}
```

L'instrumentation ne se limite pas au pilotage de l'adaptation. Elle sert aussi à l'observation de l'utilisateur dans ses interactions.

MODELE UTILISATEUR

De façon générale, le modèle utilisateur vise l'adaptation du système interactif à l'utilisateur final dans un souci de confort, d'efficacité et de sécurité [21]. Les modèles utilisateurs ont été particulièrement exploités dans les systèmes d'aide intelligents [7] et les interfaces adaptatives [24]. Le modèle utilisateur contient toute la connaissance que le système a de l'utilisateur. Les attributs typiquement maintenus dans le modèle sont les préférences, intérêts, buts et besoins d'un utilisateur ainsi que l'historique de l'interaction (tâches en cours, tâches terminées, erreurs, etc.). Pour nos besoins, afin de prendre en compte les changements de contexte pouvant survenir, il convient d'inclure également des informations relatives au contexte d'usage (contexte courant et historique). Toutes ces informations peuvent être collectées par le système à l'exécution et reportées dans le modèle utilisateur. Parmi les techniques existantes pour construire et exploiter un modèle utilisateur, citons les techniques bayésiennes [20], les techniques basées sur l'inférence, ou encore l'intelligence artificielle (réseaux de neurones). La programmation par l'exemple peut aussi permet-

tre d'inférer, à partir d'observations, les buts et besoins de l'utilisateur [13].

La mise en œuvre des actions peut être conditionnée par une validation explicite de l'utilisateur. Typiquement, la distribution d'une tâche vers une seconde plate-forme peut être placée sous contrôle explicite de l'utilisateur, critère ergonomique préconisé dans le référentiel de Bastien et Scapin [3]. Il s'agit de mettre l'utilisateur final dans la boucle. C'est l'objet de la méta-IHM. La méta-IHM de PlasticSMS (Figure 1) place les redistributions sous le contrôle de l'utilisateur, libre à lui de les ordonner, les accepter ou les refuser. Ses préférences alimentent le modèle utilisateur. Le choix de placer l'utilisateur dans la boucle relève d'une stratégie qui peut aussi être décrite par une règle d'adaptation :

AR12: `if` action=Tx/Py `do` negotiate action

Les stratégies sont prises en compte dans le moteur d'adaptation. Les choix de l'utilisateur sont collectés et constituent une base d'observations dont l'analyse permet d'inférer les préférences de l'utilisateur et, si nécessaire, d'amender les règles d'adaptation. Dans PlasticSMS, le modèle utilisateur porte sur les préférences de l'utilisateur en termes de distribution. Plusieurs cas sont envisageables. Soit l'utilisateur apprécie toujours la possibilité de distribution, auquel cas on pourrait inhiber la stratégie de négociation prônée par la règle AR12. Soit il n'apprécie pas toujours cette distribution, on pourrait alors modifier AR12 en conséquence de façon à interdire toute migration. Enfin, il peut être impossible de se prononcer suite à des contradictions dans les actions de l'utilisateur. Les contradictions peuvent n'être qu'apparentes, liées à une incomplétude du modèle. On touche ici à la modélisation imparfaite du contexte [19]. Dans PlasticSMS, par exemple, nous ne couvrons pas l'aspect social du contexte d'usage. Or, l'utilisateur seul à la maison pourra vouloir lire son SMS depuis sa télévision alors qu'il s'y refusera en présence d'autres personnes.

CONCLUSION ET PERSPECTIVES

La plasticité des IHM est une propriété aujourd'hui massivement traitée en termes de description d'IHM. Dans cet article, l'accent est porté sur les règles d'adaptation. Nous en proposons un métamodèle servant deux objectifs : (1) aider le concepteur à se poser les bonnes questions ; (2) faciliter, dans une approche itérative, l'implémentation de l'adaptation. La vision est une instrumentation automatique du code par insertion de points d'observation et d'action : observation des changements de contexte d'usage, adaptation du système interactif (action), observation de l'utilisateur dans l'interaction, révision des règles d'adaptation (action). Les perspectives sont en conséquence nombreuses. Nous avons opté pour un prototype en largeur (applicatif PlasticSMS simple mais à couverture fonctionnelle de l'adaptation com-

plète). Il s'agira ensuite de travailler en profondeur et de se confronter en particulier au problème de la complexité.

BIBLIOGRAPHIE

1. ATL : Atlas Transformation Language. <http://www.eclipse.org/m2m/atl>.
2. Balme, L., Demeure, A., Barralon, N., Coutaz, J., Calvary, G. *CAMELEON-RT: a Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces*. In Proceedings of the 2nd European Symposium on Ambient Intelligence, EUSAI 2004, Lecture Notes in Computer Science, Vol. 3295, Markopoulos et al. (Eds), Springer-Verlag Heidelberg (Publ.), Eindhoven, The Netherlands, November 2004, pp. 291-302.
3. Bastien, J.M.C., Scapin, D. *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*. Rapport technique INRIA, N°156, Juin 1993.
4. Beaudoin-Lafon, M. *Designing Interaction, not Interfaces*. In Proceedings of the Working Conference on Advanced Visual Interfaces, May 25-28, Gallipoli, Italy, 2004.
5. Berti, S., Paternò, F. *Migratory MultiModal interfaces in MultiDevice Environments*. In Proceedings of the 7th International Conference on Multimodal Interfaces, ICMI 2005, ACM Publ., pp. 92-99.
6. Bluetooth®. <https://www.bluetooth.org>.
7. Brusilovsky, P., Schwarz, E. *User as Student: Towards an Adaptive Interface for Advanced Web-Based Applications*. In A. Jameson, C. Paris, and C. Tasso (Eds), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pp. 177-188.
8. Calvary, G., Coutaz, J., Thevenin, D. *Supporting Context Changes for Plastic User Interfaces: a Process and a Mechanism*. In Proceedings of HCI-IHM 2001, A. Blandford, J. Vanderdonck, P. Gray Eds., BCS Conference series, Springer Publ., pp. 349-363.
9. Calvary, G., Coutaz, J. *Plasticité des Interfaces : une nécessité!* Information-Interaction-Intelligence, Actes des deuxièmes Assises nationales du GDR I3, Cépaduès Editions, J. Le Maître (Ed), Nancy, France, Décembre 2002, pp. 247-261.
10. Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., Young, R.M. *Four easy pieces for assessing the usability of multimodal interaction: the CARE properties*. In Proceedings of INTERACT'95, pp. 115-120.
11. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D. *Context is Key*. Communication of the ACM (CACM), Volume 48, Issue 3, ACM Press, March 2005, pp. 49-53.
12. Coutaz, J. *Meta-User Interfaces for Ambient Spaces*. Invited talk at Tamodia'06, Hasselt, Belgium, October 2006.
13. Cypher, A. *Eager: Programming Repetitive Tasks by Demonstration*. In Cypher A. (Ed.) "Watch What I Do: Programming by Demonstration", MIT Press, Cambridge MA, 1993, pp. 205-217.
14. Dâassi, O. *Les comets : une nouvelle generation d'Interacteurs pour la Plasticité des Interfaces Homme-Machine*. Thèse de Doctorat de l'Université Joseph Fourier, Grenoble I, Janvier 2007.
15. Demumieux, R., Habbouche, F. *Facilité d'utilisation et usages des MMS sur les mobiles*. Mobilité & Ubiquité '04, Nice, France, Juin 2004.
16. Dittrich, K.R., Gatzui, S., Geppert, A. *The Active Database Management System Manifesto: A Rule-base of ADBMS Features*. In Proceedings of the 2nd International Workshop on Rules in Database Systems, Vol. 985, Springer-Verlag, 1995, pp. 3-20.
17. Florins, M. *Graceful Degradation of User Interfaces*. Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in Management Sciences, Université catholique de Louvain, Belgique, 2006.
18. Hariri, M.A., Tabary, D., Kolski, C. *Plastic HCI generation from its abstract model*. In Proceedings of IASTED-HCI International Conference on Human-Computer Interaction, November 14-16, 2005, Phoenix, USA, ACTA Press, Anaheim, USA, pp. 246-251.
19. Henriksen, K., Indulska, J. *Modelling and Using Imperfect Context Information*. In Workshop on Context Modeling and Reasoning (CoMoRea), 2nd IEEE Conference on Pervasive Computing and Communications (PerCom), Orlando, Florida, USA, March 2004, pp. 33-37.
20. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, K. *The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users*. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, July 1998, pp. 256-265.
21. Johansson, P. *User Modeling in Dialog Systems*. Santa Anna IT Research Institute Report: SAR 02-2, 2002.

22. Lachenal, C. *Modèle et infrastructure logicielle pour l'interaction multi-instrument multisurface*. Thèse de Doctorat de l'Université Joseph Fourier, Grenoble I, Décembre 2004, 200 p.
23. Limbourg, Q. *Multi-Path Development for User Interfaces*. Dissertation for the degree of Doctor in Philosophy in Management Sciences, Université catholique du Louvain, 2004.
24. Liu, J., Wong, C.H., Hui, K.K. *An Adaptive User Interface Based On Personalized Learning*. In IEEE Intelligent Systems 18 (2) 2003, pp. 52-57.
25. Maragoudakis, M., Tselios, N.K., Fakotakis, N., Avouris, N.M. *Improving SMS Usability Using Bayesian Networks*. In Proceedings of the 2nd Hellenic Conference on Artificial Intelligence: Methods and Applications of Artificial Intelligence, April 2002, pp. 179-190.
26. Sottet, J.S., Calvary, G., Favre, J.M. *Ingénierie de l'Interaction Homme-Machine Dirigée par les Modèles*. Actes des Premières Journées sur l'Ingénierie Dirigée par les Modèles, IDM'05, Paris, Juillet 2005, pp. 67-82.
27. Stuerzlinger, W., Chapuis, O., Phillips, D., Roussel, N. *User Interface Façades: Towards Fully Adaptable User Interfaces*. In Proceedings of UIST'06, the 19th ACM Symposium on User Interface Software and Technology, October 2006, ACM Press, pp. 309-318.
28. Tegic Communications, T9 Text Input. <http://www.t9.com>.
29. Thevenin, D., Coutaz, J. *Plasticity of User Interfaces: Framework and Research Agenda*. In Proceedings of the Seventh IFIP International Conference on Human-Computer Interaction, INTERACT'99, Edinburgh, Scotland, 1999, pp. 110-117.
30. Vanderdonckt, J., Bouillon, L., Souchon, N. *Flexible Reverse Engineering of Web Pages with VAQUITA*. In Proceedings of IEEE 8th Working Conference on Reverse Engineering, WCRE'2001, IEEE Press, pp. 241-248.
31. Weiser, M. *The Computer for the 21st Century*. In Scientific American, 265(3), 1991, pp. 1613-1619.
32. Wu, H., de Kort, E., De Bra, P. *Design Issues for General-Purpose Adaptive Hypermedia Systems*. In Proceedings of the ACM Conference on Hypertext and Hypermedia, New York, 2001, ACM Press, pp. 141-150.