

---

# Principes et étude de cas d'adaptation d'IHM dans les SI en fonction du contexte d'interaction de l'utilisateur

Anas Hariri<sup>\*\*</sup>, Sophie Lepreux<sup>\*</sup>, Dimitri Tabary<sup>\*</sup>, Christophe Kolski<sup>\*</sup>

<sup>\*</sup> Université de Valenciennes

LAMIH –CNRS UMR 8530, Le Mont-Houy  
59313 Valenciennes cedex 9, France

{Prénom.Nom}@univ-valenciennes.fr

<sup>\*\*</sup> Université de Concordia

1515 St. Catherine Ouest, EV009.189  
Montréal, QC H3G2W1, Canada

ahariri@encs.concordia.ca

---

**RÉSUMÉ.** La mobilité a engendré de nouveaux besoins informatiques. Les téléphones portables, assistants personnels et autres moyens de communications sont désormais utilisés pour l'ensemble des tâches effectuées par les utilisateurs. Les utilisateurs souhaitent utiliser les services spécifiques liés à cette mobilité ainsi que les applications habituelles de leur lieu de travail. Les interfaces utilisateur doivent par conséquent s'adapter au contexte d'interaction de l'utilisateur. Cet article décrit une démarche de conception de systèmes d'information équipés d'interfaces plastiques. Cette démarche permet de générer l'interface utilisateur et de l'adapter en fonction du contexte, en respectant l'utilisabilité. Les patrons sont utilisés pour relier le problème d'adaptation de l'interface, le contexte et les solutions qui peuvent être utilisées dans le but de mieux prendre en compte l'utilisateur. L'étude de cas montrant l'accès d'un système d'information de supervision par un utilisateur nomade permet d'illustrer la démarche.

**ABSTRACT.** With the mobility, new needs have been found. From now on, the cellular phone, the personal assistant and other modern communication means are used by users for realizing their tasks. The users wish using services specific to this mobility and also applications used usually at office. The user interface has to be adaptable to the interaction context of user. This paper describes an approach of design of information systems equipped with adaptable UI. Our approach allows generating UI and adapting it dynamically to the context while preserving its usability. UI design patterns are used to connect UI adaptation problems and the context of use to usable design solutions in order to better taking into account the user. The case study showing access to a supervision information system by a nomadic user allows illustrating our approach.

**MOTS-CLÉS :** conception d'IHM, interface homme-machine plastique, patrons, prise en compte de l'utilisateur, contexte, adaptation.

**KEYWORDS:** UI design, adaptable user interface, patterns, user-centered design, context, adaptation.

---

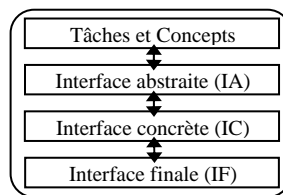
## 1. Introduction

Faisant l'objet d'un vaste courant de recherche au niveau international, l'informatique pervasive laisse envisager une nouvelle génération de systèmes d'information (Henricksen *et al.*, 2004), et nécessite de nouvelles modalités d'interaction homme-machine permettant de prendre en compte le contexte d'interaction de l'utilisateur. Schilit et Theimer sont parmi les premiers à avoir décrit un système d'information (SI) sensible au contexte (qualifié de Context-Aware (Schilit *et al.*, 1994)) ; ce système utilisait des informations sur la localisation et l'identité des acteurs à proximité de l'utilisateur, et prenait en compte des changements liés aux informations utilisées. Les travaux suivants (Brown *et al.*, 1997), dans ce domaine, utilisaient aussi d'autres informations concernant l'environnement et le niveau intellectuel de l'utilisateur afin de fournir des informations et des services à celui-ci.

Des recherches actives portent sur de nouveaux types d'IHM, qu'elles soient dites sensibles au contexte, ou encore plastiques. Les recherches présentées dans cet article se focalisent sur de telles interfaces. Elles disposent de la capacité de s'adapter de manière dynamique à un changement de contexte tout en préservant des propriétés ergonomiques prédéfinies (Grolaux *et al.*, 2001). Elles doivent avoir la capacité de détecter et utiliser des informations contextuelles, telles que la localisation de l'utilisateur, la date et l'heure, la proximité d'autres utilisateurs, les ressources disponibles, les caractéristiques de la plate-forme cible, les conditions ambiantes, etc. Nous souhaitons, durant l'exécution, répondre dynamiquement aux modifications contextuelles dans le respect de l'utilisabilité (Nielsen, 1993). La norme ISO 9241-11 définit l'utilisabilité de la manière suivante : « un système est utilisable lorsqu'il permet à l'utilisateur de réaliser sa tâche avec efficacité, efficience et satisfaction dans le

contexte d'utilisation spécifié ». Le contexte d'utilisation, dit aussi contexte d'usage, concerne la plate-forme, l'environnement et l'utilisateur. Il est la base pour mieux prendre en compte l'utilisateur dans l'utilisation du SI.

L'adaptation des IHM à leur contexte d'usage est devenue un sujet d'actualité. Notons particulièrement les recherches menées sur le générateur d'IHM plastiques *ArtStudio* (Thevenin *et al.*, 1999) et les *Comets* (Demeure *et al.*, 2006), qui sont des interacteurs dédiés à la plasticité et qui bénéficient de représentations multiples. La majeure partie des travaux, portant sur la plasticité de l'IHM (Calvary *et al.*, 2003), (Thevenin *et al.*, 1999), prend en compte généralement une gamme limitée de contextes d'usage, se focalisant sur des caractéristiques comme la taille de l'écran, la langue... Ceci est dû à la difficulté de capturer et de traiter les informations contextuelles. Dans l'espace problème de la plasticité, Calvary *et al.* (2002a) envisagent les trois leviers de la réaction au changement de contexte : une action sur le contexte d'usage ; un remodelage de l'interface ; ou une redistribution de l'interface sur les différentes plates-formes composant l'environnement. La majorité des recherches vise à manipuler l'interface utilisateur sans se préoccuper de la partie fonctionnelle. A l'opposé, nous retiendrons le projet *Rainbow* (Garlan *et al.*, 2004) qui utilise une infrastructure réutilisable et vise une adaptation dynamique (à l'exécution) en plus d'une prise en compte du processus métier du SI. La migration de l'IHM pendant l'exécution entre les plates-formes avec la conservation des données et des informations de l'utilisateur reste un problème difficile. Ce problème a notamment été étudié par (Bandelloni *et al.*, 2004) dans le cadre du projet *Cameleon*. Notre démarche suivra les étapes du cadre *Cameleon*, qui définit quatre étapes essentielles de développement des interfaces dans un environnement pervasif (cf. Figure 1) : tâches et concepts, interface abstraite, interface concrète, et interface finale.



**Figure 1.** *Quatre principales étapes de développement du projet Cameleon*

L'objectif de nos travaux consiste à générer un SI possédant une IHM plastique telle qu'elle est définie précédemment à partir de son modèle abstrait. Nos précédents travaux se sont focalisés sur la spécification et la génération d'interfaces multiplateformes (Hariri *et al.*, 2005). Nous avons proposé une démarche complète de génération d'une IHM plastique (Hariri *et al.*, 2007, 2008) qui a ensuite fait l'objet d'une thèse (Hariri, 2008). La section 2 rappellera ses points importants. Les concepts indispensables et innovants dans la prise en compte de l'utilisateur et son contexte d'interaction sont détaillés dans un premier temps et la démarche est présentée dans un second temps. La section 3 propose un retour sur expérience permettant d'illustrer la démarche d'une part et de valider la prise en compte de l'utilisateur d'autre part. L'étude de cas porte sur la mise en place d'un SI de supervision et son utilisation par des utilisateurs nomades. Enfin, cet article se termine par une conclusion et des perspectives de recherche.

## 2. Démarche pour la génération d'IHM plastique

La démarche proposée consiste à générer automatiquement l'interface plastique du SI en utilisant la technique des patrons et en intégrant une architecture basée sur les composants métier logiciels. L'utilisation de composants métier vise à faciliter et augmenter la capacité de l'adaptation dynamique en fonction du contexte d'usage. Dans cette démarche, l'IHM est considérée comme un assemblage de composants de présentation au service des composants métier. Les composants métier ont pour but de changer dynamiquement les composants de présentation afin de s'adapter au nouvel état du contexte d'usage en évitant un retour à la conception.

### 2.1 Les Concepts clés de la démarche

#### 2.1.1 Le Contexte d'usage

Le contexte d'usage du SI contient des informations à considérer pendant la conception et l'exécution du système. Dans ce domaine trois catégories d'informations contextuelles peuvent être distinguées (Dey, 2000) : (1) celles relatives à la plate-forme, telles que ses caractéristiques (processeur, mémoire, équipements périphériques), les réseaux de connexion, la taille de l'écran d'affichage, et les outils d'interaction disponibles ; (2) celles relatives à l'utilisateur, telles que son profil, son niveau professionnel, sa localisation physique, son activité courante (loisir, travail), etc. ; (3) les informations environnementales comme le niveau de bruit, la luminosité, la température, et les informations temporelles. Ces informations permettent de prendre en compte l'utilisateur par la connaissance de ses caractéristiques, mais aussi son environnement et ses supports d'interaction. Pour nous, ce contexte d'usage est la base de la prise en compte de l'utilisateur dans le développement du SI et des techniques d'interaction. Les informations contextuelles sont stockées et actualisées lors de la capture du contexte. Celles-ci sont envoyées régulièrement au SI pour évaluer l'utilisabilité de l'IHM (Gellner *et al.*, 2001). En cas de changement au niveau du contexte, l'utilisabilité est évaluée et une adaptation est effectuée en cas de nécessité, afin de la préserver.

#### 2.1.2 Composants métier

L'approche « composant » a été introduite dans l'objectif d'améliorer la réutilisation. Elle consiste à segmenter, rationaliser, encapsuler et plus généralement modulariser les SI (Barbier *et al.*, 2002). Les composants peuvent être distingués en fonction de

leur niveau d'application dans le cycle de développement (Cauvet *et al.*, 1999). Les composants de domaine expriment des connaissances réutilisables dans le développement de tous les SI d'un domaine d'application. Ces composants sont utilisables et réutilisables dès la phase d'analyse car ils expriment souvent une connaissance de haut niveau, orientée sur la définition de problèmes et de besoins. Les composants métier se retrouvent dans cette catégorie car ils couvrent les besoins du domaine (Hassine *et al.*, 2002). Une classification à laquelle sont soumis les composants métier est celle des domaines. Wartik et Prieto-Diaz dans (Vestal, 1998) donnent les fondements de la réutilisation de connaissances de domaine. La spécification des composants pour un domaine, la flexibilité et la généralité des composants pour des usages non forcément prévus, et l'intégration et la réutilisation des composants sont des défis dans l'ingénierie de la réutilisation.

Les composants logiciels permettent de réutiliser les fonctionnalités de base en programmation (ex : les tris). Le découpage des applications en composants logiciels assemblés remet en question la construction de l'IHM globale de l'application. L'IHM est composée des composants de présentation. Ces derniers font le lien entre le noyau fonctionnel du système et l'utilisateur. Un composant de présentation a un espace d'interaction adapté aux tâches associées au composant. Il contient des informations concrètes et/ou abstraites concernant la modalité d'interaction avec l'utilisateur. Cet espace est spécifié et caractérisé en fonction de la modalité, en particulier en fonction de la plate-forme cible. Les composants de présentation peuvent être réorganisés sur le canevas de l'IHM, cachés ou affichés ; de plus les composants peuvent être reconfigurés (changement de taille, de police, de couleur, etc.) afin de s'adapter au contexte d'usage.

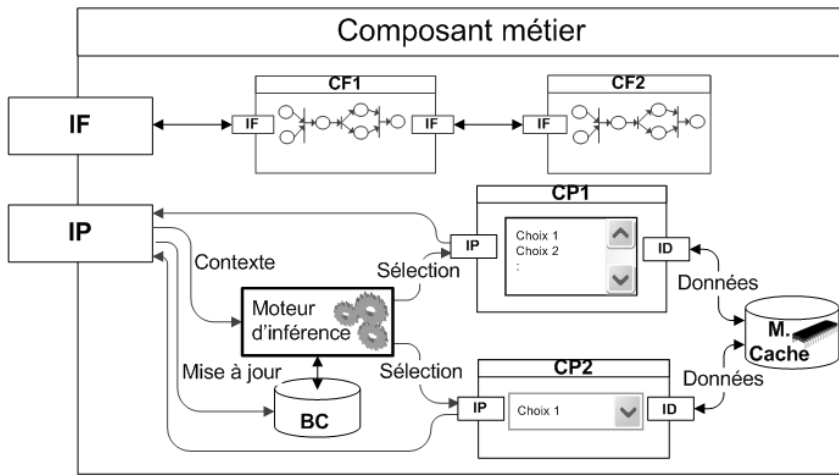
Dans le cadre de ces travaux, un composant métier est composé d'un ou plusieurs composant(s) fonctionnel(s) (contenant le noyau fonctionnel) et d'un ou plusieurs composant(s) de présentation correspondant à l'IHM. Il peut également posséder une base de connaissance intégrée dans l'architecture des composants. Nous avons proposé l'idée d'intégration d'une base de connaissance afin d'aider à prendre des décisions à l'exécution, en particulier les décisions de choix et de changement de la facette de présentation (celle-ci pouvant être constituée d'un ou plusieurs composant(s) de présentation), face au changement du contexte d'usage. L'intégration de la base de connaissance vise à : diminuer le temps d'adaptation à l'exécution (sans aucune intervention de l'utilisateur ou du concepteur au niveau des décisions à prendre) ; réduire la charge du réseau de connexion (dans la mesure où la charge peut dépendre du rechargement du système à partir du serveur, à chaque adaptation effectuée à la conception) ; optimiser le temps de réponse vis-à-vis du changement contextuel (le temps à attendre entre le moment du changement du contexte et l'application de la réaction). La Figure 2 illustre un exemple de composant métier « intelligent » (en intégrant une base de connaissance) ; appelé "*choix*", il possède :

- une interface d'entrée/sortie fonctionnelle (IF), une autre pour la présentation (IP) du composant, une base de connaissance permettant de choisir sa facette de présentation en fonction du contexte ;
- une mémoire cache<sup>1</sup> qui sera utilisée par les composants de présentation pour stocker les informations et les entrées de l'utilisateur ;
- deux composants fonctionnels *CF1* et *CF2*. Les deux composants peuvent travailler en collaboration entre eux, ou tous les deux peuvent proposer la même fonctionnalité, mais dans deux contextes différents, en prenant en compte par exemple la puissance de traitement de la machine ;
- deux composants de présentation *CPI* (de type *ComboBox*) et *CP2* (de type *ListBox*) correspondants à deux contextes différents (*C1*, *C2*). Pendant l'adaptation, un champ de présentation spécifié pour le composant métier "*choix*" et contenant une liste d'éléments, peut être projeté vers *CPI* ou *CP2* selon le contexte ;
- et une base de connaissance (BC) qui sera utilisée par un moteur d'inférence. Ce dernier donne la décision de choix de la présentation. Cette base peut être améliorée grâce à l'apprentissage (Hariri, 2008), afin de permettre de prendre en compte des nouveaux contextes.

Cette contribution est proche de celle de l'approche *Comet* (citée précédemment) qui embarque plusieurs présentations décrites, en conception, à tout niveau d'abstraction ; ensuite le choix de stratégies et politiques d'adaptation s'effectue à l'exécution. Cependant l'approche *Comet* ne propose pas de mécanismes d'apprentissage permettant d'améliorer les connaissances prédéfinies.

---

<sup>1</sup> La mémoire cache des composants métier est un espace de stockage dans lequel des informations et des entrées de l'utilisateur peuvent être sauvegardées ; elles peuvent être transférées vers une nouvelle plate-forme, puis récupérées (notion de migration d'IHM (Bandelloni et Paternò, 2004)).



**CP1, CP2** : Composants de présentation ;  
**CF1, CF2** : Composants fonctionnels ;  
**IF** : Interface fonctionnelle ;  
**IP** : Interface de présentation ;  
**BC** : Base de connaissance.

Figure 2. Exemple de composant métier

### 2.1.3 Patrons

La notion de patron est née d'un architecte (Alexander *et al.*, 1977) pour qui « chaque patron décrit un problème qui se manifeste constamment dans notre environnement et décrit l'architecture de la solution à ce problème, d'une façon telle que l'on puisse réutiliser cette solution des millions de fois sans jamais l'adapter deux fois de la même manière. ». Le patron proposé pour accompagner notre démarche a la particularité de prendre en compte le contexte d'usage (Plate-forme, Environnement et Utilisateur) qui sera utilisé pour l'adaptation. Il comprend un certain nombre de champs qui seront renseignés pour chaque patron (id, nom, description, contexte d'utilisation du patron, contexte d'usage, déclenchement, solution et collaboration avec d'autres patrons). Chaque patron sera transcrit sous la forme d'un fichier XML de manière à faciliter le stockage et la transmission de l'information. Notons que dans le cadre de ces travaux, une maquette d'outil, appelé *PaDev*, de développement de patrons utilisables dans notre démarche, a été développée (Hariri, 2008). La Figure 3 illustre un exemple de patron applicable à l'exécution. Ce patron permet de recalculer les tailles et les localisations optimales pour l'IHM, suite à un changement dans le contexte de la plate-forme (en particulier la taille de l'écran d'affichage). Il essaye de trouver les meilleures tailles et localisations pour les fenêtres et leurs composants de présentation, sous la condition que le taux du changement de la taille de l'écran varie entre 0.5 et 1.3. Si le changement est de moins de 0.5, le patron appelle le patron P10225 qui est le responsable de l'action consistant à diviser les fenêtres en sous-fenêtres afin de bien investir l'espace d'affichage disponible. Sinon, il appelle le patron P10226 qui trouve des solutions de manière à fusionner des fenêtres (Lepreux *et al.*, 2006).

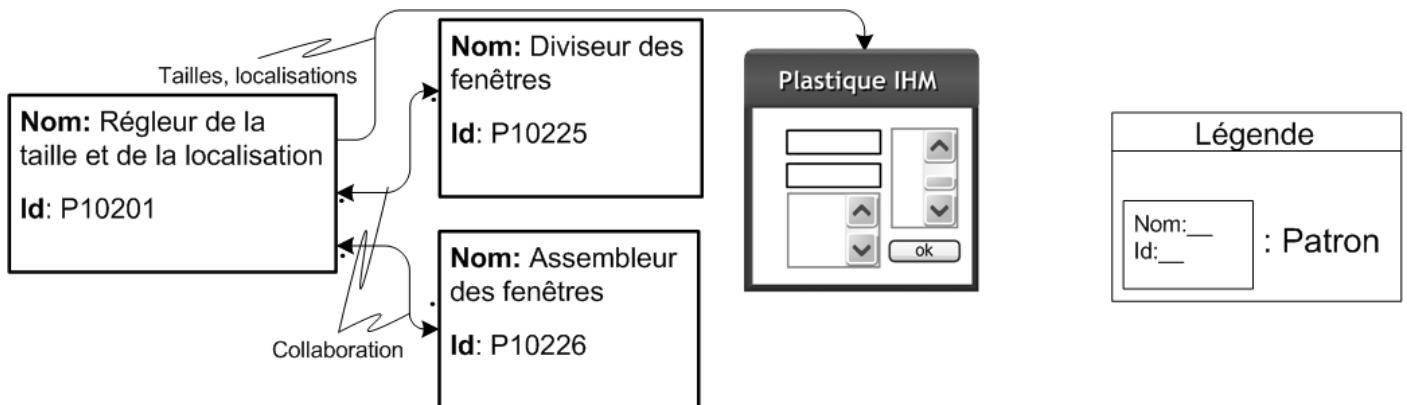


Figure 3. Exemple de patrons

## 2.2 Démarche de conception centrée sur l'utilisateur et son contexte d'interaction

La démarche proposée dans cet article est montrée en Figure 4. Elle illustre les trois niveaux principaux de la démarche de génération de l'IHM plastique : abstrait, concret et final, tels que ceux du projet Cameleon (Calvary *et al.*, 2002b).

**Le niveau abstrait** correspond à celui de l'interface abstraite, l'AMXML (Abstract Model in XML) est construit à partir d'un MIA (Modèle d'Interface Abstraite) et/ou d'un modèle de tâche. AMXML décrit d'une part l'espace d'interaction comprenant toutes les tâches à réaliser par le système, par l'utilisateur ou par l'interaction des deux ; d'autre part il peut inclure, en fonction de la richesse du MIA d'origine, des définitions du contexte d'usage à considérer pendant la conception. L'AMXML est un MIA en XML qui respecte la DTD (Document Type Definition) d'UsiXML (Limbourg *et al.*, 2005) avec deux ajouts : nous avons ajouté deux parties permettant d'intégrer le modèle de tâches dans l'AMXML, la première partie est spécifique aux tâches, et l'autre permet de définir les relations entre les tâches. L'objectif, dans cette étape, est de transcrire un modèle abstrait en AMXML. Ce

niveau inclut une bibliothèque de vocabulaires prédéfinis pour différentes représentations et une DTD à utiliser pour construire l'AMXML.

La réification de l'IHM - passage au **niveau concret** (dépendant de la modalité)- passe par la construction de la structure métier et une première adaptation avant la distribution du SI à la plate-forme cible. Ce niveau vise la conception d'un SI ayant une interface plastique adaptée au contexte prévu. Ce niveau est équipé d'une bibliothèque de composants métier et de deux autres de patrons. La première bibliothèque de patrons concerne le problème de choix des composants métier. Elle fournit au système des patrons qui ont pour but d'aider à choisir les composants métier convenables à un modèle de tâche. Ceux-ci préconisent aussi des solutions de compositions. Ces patrons permettent de prendre en compte l'utilisateur par le biais des tâches qu'il doit réaliser. La seconde bibliothèque de patrons (partagée avec le niveau final de la démarche) vise le problème du choix et de l'utilisation de composants de présentation. Elle fournit des patrons qui préconisent des solutions pour choisir et assembler les composants de présentation pendant l'adaptation primaire (adaptation appliquée à la conception) mais aussi l'adaptation vivante (adaptation effectuée à l'exécution du SI). Ces patrons permettent de prendre en compte l'utilisateur par le biais du contexte d'usage prévu ou constaté.

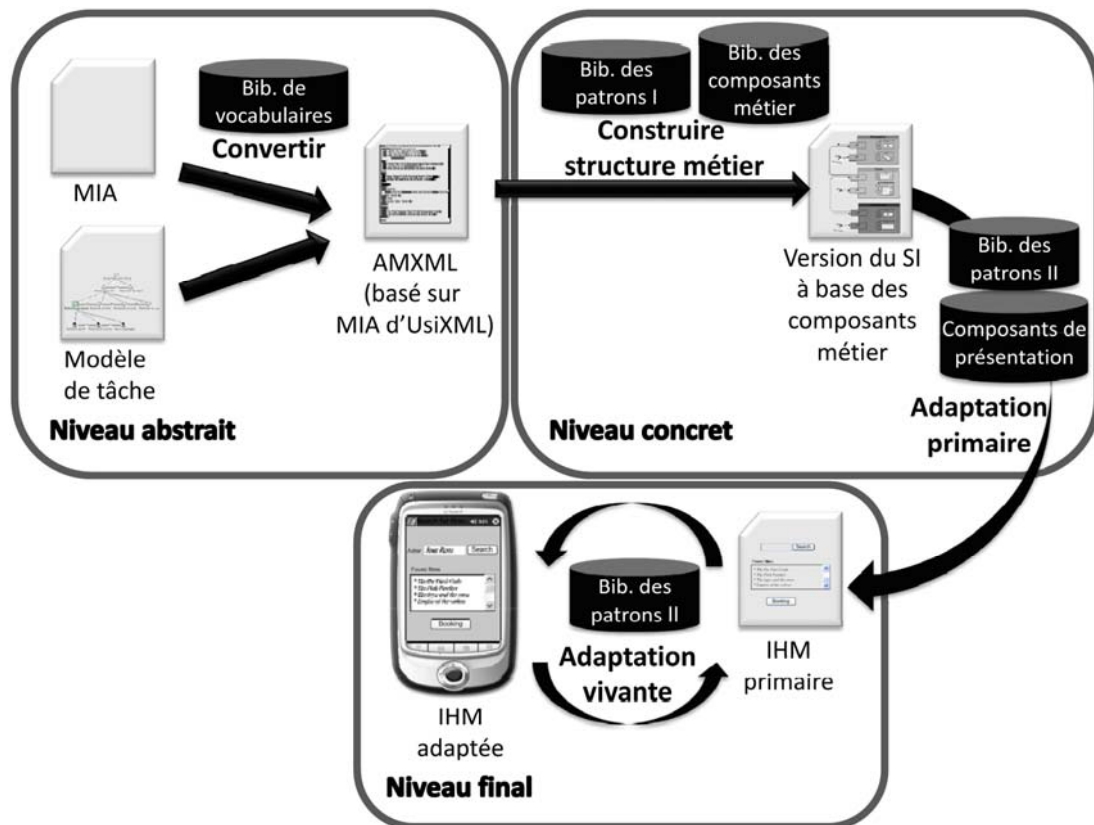


Figure 4. Démarche globale pour la génération d'interface plastique

Au **niveau final**, le SI et son IHM plastique seront utilisés. Des processus de capture du contexte et d'apprentissage coopèrent pour faire évoluer les bases de connaissance du système afin de conserver l'utilisabilité de l'IHM de manière à ce qu'elle soit en permanence adaptée à l'utilisateur. L'adaptation a pour objectif de préserver l'utilisabilité de l'IHM, alors que l'apprentissage devrait permettre de préserver l'utilité de l'adaptation.

### 2.2.1 Conversion du MIA en AMXML

La conversion débute par l'identification de la représentation du MIA (ex. : le MIA respecte le formalisme de la méthode de spécification et conception d'IHM TOOD (Tabary, 2001), ou celui de DIANE+, etc.) ; l'identification permet de choisir un vocabulaire à utiliser pour traduire le MIA en AMXML. La traduction s'appuie sur une DTD (détaillée dans (Hariri, 2008)) et sur une bibliothèque de vocabulaires assurant les correspondances du MIA d'origine vers l'AMXML. Nous proposons dans (Hariri, 2008) des règles permettant de construire l'AMXML à partir du modèle d'interface abstraite et/ou du modèle de tâche.

### 2.2.2 Construction de la structure métier

L'objectif de cette étape est de construire la structure métier de l'IHM. Tout d'abord, une liste de patrons est choisie pour le choix des composants métier et leur assemblage. Le choix des patrons est effectué selon les tâches et le domaine d'application du SI, spécifiés dans le MIA. Deux grandes catégories de tâches peuvent être distinguées : (1) les tâches interactives effectuées par l'utilisateur et (2) les tâches métier dépendantes du domaine. Chacun des composants métier correspond à une ou plusieurs tâche(s). Afin de construire la structure métier, tout d'abord, une liste des patrons est choisie. Le choix des patrons est basé sur un SIAD (Système Interactif d'Aide à la Décision) considérant les contraintes du domaine et des tâches abstraites du système. La connaissance est déterminée par le concepteur pendant le développement des patrons. Ensuite, des composants métier sont

sélectionnés, en respectant les patrons, parmi une bibliothèque de composants métier classifiés par domaine d'utilisation. La sélection s'effectue selon les tâches spécifiées potentiellement dans le MIA. Enfin, les composants sélectionnés sont configurés et reliés à l'aide des patrons. Une « glue » (Bayle, 1998), adaptée au middleware qui impose une interface de communication, est spécifiée et adaptée au besoin des interfaces de communication des composants métier. Ces interfaces logicielles permettent d'accéder aux méthodes et attributs fournis par les composants. La Figure 5 illustre une structure métier possible pour trois tâches utilisateurs (*Copier*, *Couper*, et *Coller*). La structure métier est composée de trois composants métier nécessaires pour effectuer les trois fonctionnalités.

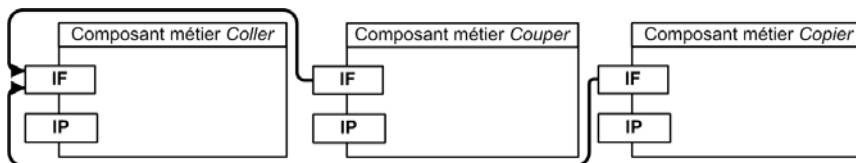


Figure 5. Exemple de structure métier

### 2.2.3 Adaptation de l'IHM primaire au contexte d'interaction

La dernière étape du niveau concret consiste à produire l'IHM. De la même manière que pour le choix des composants fonctionnels, des patrons sont choisis dans la bibliothèque en fonction du contexte d'usage et de la tâche. Ils seront utilisés pour la sélection des composants de présentation. Rappelons que le composant métier possède, éventuellement, plusieurs facettes de présentation associées à différents contextes. Les composants de présentation adéquats au contexte d'usage prévu sont sélectionnés à l'aide des patrons choisis. Par exemple, dans le cas d'une modalité graphique, un composant de présentation avec une liste de choix peut afficher à l'utilisateur, dans son espace d'interaction, une liste contenant un ensemble de choix, l'utilisateur pouvant alors en choisir un. Alors qu'avec une modalité vocale, le composant de présentation donne à chaque élément de choix un numéro puis attend l'entrée de l'utilisateur pour détecter son choix. Les composants de présentation sélectionnés sont assemblés et configurés. Les facettes de présentation sont arrangées grâce à un guide de style qui offre des spécifications et des informations supplémentaires sur le dispositif cible. A ce moment, le code d'une version primaire du SI avec IHM plastique est généré et distribué sur la plate-forme.

### 2.2.4 Utilisation du SI et adaptation au contexte d'interaction lors de l'exécution

Le SI est exploité sur la plate-forme cible. Les tâches sont effectuées par les composants fonctionnels. Le contexte d'usage est actualisé avec des informations concernant l'utilisateur, la plate-forme et l'environnement. L'étape d'adaptation vivante (à l'exécution) est appelée lors d'un changement contextuel. En cas d'action ou de changement de contexte, l'utilisabilité du SI est évaluée afin de provoquer la réadaptation. Les critères d'utilisabilité intègrent la qualité du dialogue qui peut être mesurée par des propriétés du dialogue. Il s'agit d'un ensemble de propriétés centrées utilisateur. Pendant l'utilisation du SI, les réactions de l'utilisateur aux changements de l'IHM, en fonction de l'adaptation, sont capturées et envoyées au processus d'apprentissage qui les utilisera pour extraire de nouvelles connaissances, en comparant le changement au niveau contextuel et les réactions répétées de l'utilisateur. Puis les connaissances extraites sont évaluées et validées afin d'améliorer la qualité des décisions. Un arbre de décision est construit pendant la génération de l'interface. Il est ensuite amélioré par le processus d'apprentissage automatique. De plus, l'accès à la base de connaissance et les modifications au niveau des règles seront aisément effectués grâce à la représentation de l'arbre de décision en format standard de type XML. Dans le cas où l'adaptation est nécessaire durant l'exécution (cf. Figure 6), l'IHM est évaluée selon les critères d'utilisabilité pour décider le type d'adaptation nécessaire. Cette évaluation permet de savoir si l'adaptation vivante est capable de prendre en compte les changements contextuels. Si tel est le cas, une ou plusieurs parties de l'IHM qui doivent être modifiées sont détectées afin de préserver l'utilisabilité de l'IHM. Puis une liste des patrons relatifs à la sélection des composants de présentation est choisie.

Les composants de présentation sont re-sélectionnés à l'aide des bases de connaissance de leurs composants métier, adéquatement au nouveau contexte d'usage. Les composants de présentation sont réassemblés ; le code implémentable de la partie adaptée est mis à jour. Par contre si l'interface n'est plus capable de s'adapter, le SI désassemble les composants de présentation de certains composants métier. Ces composants métier sont transmis à l'adaptation primaire du niveau conception afin de régénérer totalement l'IHM.

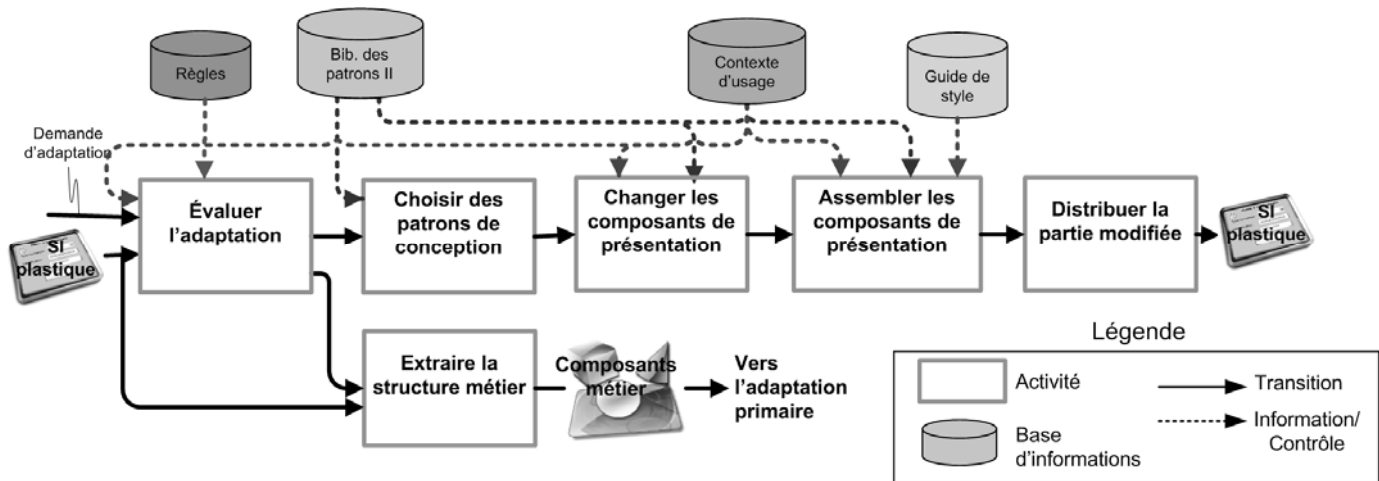


Figure 6. Processus d'adaptation vivante

### 3. Etude de cas : SI prenant en compte l'utilisateur dans un cadre de supervision d'une usine chimique

Dans cette étude de cas, nous appliquons notre démarche à un SI utilisé en supervision industrielle. Tout d'abord, nous abordons le scénario du système, avec les différentes facettes du contexte d'usage envisagé. Ensuite, des spécifications des tâches en CTT (Paterno, 1999) sont utilisées afin de construire le modèle AMXML. Puis, des composants métier sont sélectionnés et reliés. L'IHM est générée sous la forme d'une première version. A l'exécution, l'adaptation vivante s'opère.

#### 3.1 Présentation de l'application industrielle

L'application concerne le cas d'un procédé industriel de fabrication de godets métalliques remplis par une solution chimique préparée préalablement. Le scénario du système est adapté de (Riahi, 2004 ; Moussa *et al.*, 2006). Le processus simplifié comprend deux phases : (1) la première consiste à façonner les godets par emboutissage de pièces métalliques prédécoupées ; (2) la seconde consiste à verser dans les godets une dose de solution chimique. L'opération de remplissage n'est autorisée que si la température de la cuve  $T_{cuve}$  ne dépasse pas les  $55^{\circ}\text{C}$ . Il est question de supervision, de détection d'alerte, et de commande. Nous supposons que l'ensemble de ces tâches est attribué à un ou plusieurs opérateurs humains.

#### 3.2 Contextes d'usage envisagés

Nous aborderons, dans cette partie, les différentes facettes du contexte d'usage à envisager. L'IHM plastique du SI doit les prendre en compte et observer les changements contextuels à l'exécution. Nous verrons que les tâches de l'utilisateur (contexte relatif à l'utilisateur) peuvent être indiquées ou modifiées en fonction des données de contrôle, autrement dit les quantités mesurées (contexte environnemental).

##### 3.2.1 Contexte relatif à l'utilisateur

L'utilisateur du SI visé (cf. Tableau 1) peut être un opérateur en salle de contrôle qui peut intervenir directement selon les autorisations données : (1) le superviseur en chef qui administre et visualise tous les synoptiques de toutes les sections, et en informe les opérateurs responsables des processus en cas de nécessité ; (2) les experts d'astreinte qui peuvent surveiller et intervenir à distance (éventuellement de chez eux) ; (3) les opérateurs avec un profil particulier qui leur permet d'accéder à certaines informations et peuvent intervenir de manière très limitée. L'IHM doit prendre en compte la catégorie de l'utilisateur, et respecter les autorisations attribuées à chacune et celles temporaires qui peuvent être données en cas d'urgence. Certains utilisateurs travaillent en mobilité, tandis que d'autres partagent leur temps entre la salle de contrôle et le terrain. Ils peuvent être en mobilité intra-entreprise (au sein d'un site de l'entreprise) ou extra-entreprise, ils travaillent alors à distance et accèdent au SI de l'extérieur en cas de nécessité. Chaque utilisateur peut spécifier ses préférences concernant l'IHM du système (couleurs préférées, mode d'interaction favori,...). De plus, pour un utilisateur possédant un éventuel handicap, l'IHM doit pouvoir offrir des couleurs spéciales pour le fond d'écran et pour les composants de présentation, des messages sonores et/ou des alertes vibrantes selon le type de handicap.

**Tableau 1. Facettes du contexte relatif à l'utilisateur**

Type d'opérateur	Autorisations	Mobilité	Heures de travail	Mode d'interaction préféré	Couleurs préférées	handicap
Superviseur en chef	Arrêt des machines	Non	Créneau horaire	Vocal	Noir/blanc	Non
Opérateur en salle de contrôle	Réglage des machines	Nomade		Tactile	Bleu/blanc	Visuel
Opérateur d'astreinte	Intervention à distance	Intra-entreprise		Clavier	...	Auditif
Opérateur avec profil particulier		extra-entreprise		...		...

### 3.2.2 Contexte relatif à l'environnement

Les caractéristiques de l'environnement sont variées et peuvent changer pour certaines d'entre elles fréquemment (cf. Tableau 2). L'IHM doit prendre en compte une gamme large de facettes liées au contexte relatif à l'environnement. Le SI peut être utilisé dans la salle de contrôle, à l'intérieur de l'entreprise, ou dans le véhicule ou au domicile de l'opérateur d'astreinte qui peut se connecter au SI et intervenir à distance. La stabilité du niveau de luminosité est variée selon le lieu de travail. L'environnement de travail peut être très bruyant (à côté des machines, ou dans la voiture), un peu bruyant (dans la salle de contrôle si plusieurs opérateurs y travaillent dans des situations où les communications sont alors nécessaires), ou calme (bureau du superviseur en chef). Dans un environnement bruyant, le SI peut être contraint d'utiliser (ou proposer) un niveau sonore élevé avec des alertes vibrantes. En revanche, en environnement calme, il s'agit de baisser le son, utiliser des alertes vibrantes et/ou des témoins lumineux.

**Tableau 2. Facettes du contexte relatif à l'environnement**

Lieu	Luminosité	Bruit	Heure	Etat	Situation	Données de contrôle
Salle de contrôle	Basse	Bruyant	Valeur au format heure (ex : 12:00)	Urgent	Travail	35°
Section I	Moyenne	Calme		Normal	Hors-travail	...
Extérieur	Elevée			...	Occupé	

Les opérateurs nomades sont équipés de dispositifs mobiles disposant de fonctions de communication et de localisation. Cela permet de les localiser à tout moment ; par ex. : le SI peut les localiser dans un voisinage ou zone donnée, et leur envoyer des alertes basées sur la proximité de certains événements (avec localisation et type). Les données de contrôle (comme la température, l'alimentation électrique, l'état des machines,...) peuvent jouer un rôle très important, conduisant à changer les tâches ou les autorisations de l'utilisateur. Ces données doivent être représentées selon un ou plusieurs types d'échelles de mesure (Petersen et May, 2006), l'utilisateur ayant à prendre des décisions s'appuyant sur ces données de contrôle.

### 3.2.4 Contexte relatif à la plate-forme

Les opérateurs nomades, de même que ceux procédant à la surveillance des installations de l'entreprise (sans être localisés en salle de contrôle), peuvent être munis d'ordinateurs ultra-portables, de PDA, ou de téléphones portables. Les autres dans la salle de contrôle peuvent travailler sur des plates-formes associées à plusieurs écrans d'affichage. Chaque plate-forme a ses propres caractéristiques. Le processus d'adaptation doit prendre en compte les différentes caractéristiques de la plate-forme cible (cf. Tableau 3).

**Tableau 3. Facettes du contexte relatif à la plate-forme**

Type	Système d'exploitation	Connexion réseau	Système de localisation	Mode d'affichage	Espace d'affichage	Couleurs	Modes d'interaction disponibles	Style
PDA	Windows mobile	Non	Non	Horizontal	220*300	4096	Clavier	Fenêtre, Menu, ...
PC	Symbian	Wifi	GPS	Vertical	640*480	65536	Clavier/	Page Web, bouton, ...
...	Linux	...	...		...	...	Joystick	Vocal
	...						Ecran tactile	
							...	

## 3.3 Application de la démarche

Nous supposons que nous partons d'un modèle de tâche spécifié en TOOD ; les règles de construction d'AMXML seront appliquées au fur et à mesure sur ce modèle afin de générer le modèle AMXML. Pendant la construction, le concepteur peut intervenir pour spécifier, par exemple, le contexte d'usage prédéfini et les ressources de l'IHM. La structure métier sera ensuite générée. Puis à l'aide de l'expérience des patrons, les composants métier sont reliés. Une version primaire du SI sera générée, prenant en compte le contexte prévu, et sera prête à être distribuée vers le dispositif cible. L'adaptation vivante sera appelée à l'exécution lors d'un changement contextuel. L'adaptation vivante permettra de changer la présentation des quantités mesurées (température, vitesse, etc.) sans nécessité de revenir à la conception suite à un important changement dans les quantités mesurées, ou un éventuel changement du type de tâche de l'utilisateur (contexte relatif à l'utilisateur). Enfin, les tâches de l'utilisateur



peuvent être modifiées à l'exécution en fonction des valeurs des quantités mesurées (par exemple, la température  $T_{cuve}$  atteint la valeur critique).

### 3.3.1 Construction d'AMXML

Comme le système de supervision est vaste, nous nous intéresserons seulement à la tâche *Recevoir alerte* du système. Les tâches de cette partie du SI sont exprimées dans un modèle de tâche s'appuyant sur CTT (cf. Figure 7).

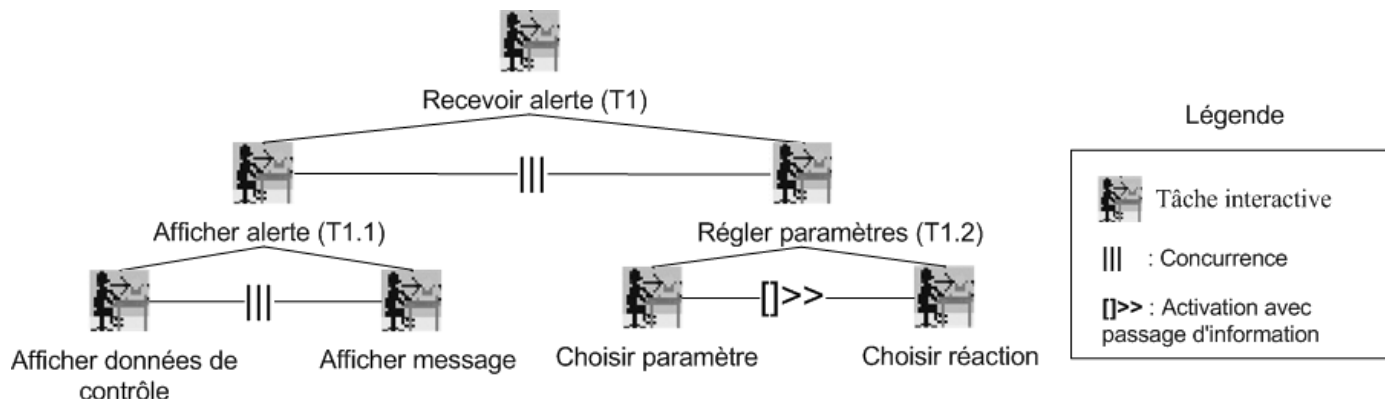


Figure 7. Réécriture en CTT du modèle de tâche "Recevoir alerte" TOOD

Tout d'abord, des *conteneurs abstraits* sont associés à la tâche principale *Recevoir alerte*, et aux sous-tâches mères *Afficher alerte*, *Régler paramètres*. Ensuite des *composants abstraits* sont associés aux tâches interactives terminales *Afficher données de contrôle*, *Afficher message*, *Choisir paramètre*, et *Choisir réaction*. Les *composants abstraits* sont placés dans les *conteneurs abstraits* de leurs tâches mères.

La partie `<taskModel>` de l'AMXML se compose des tâches du système et d'une sous-partie `<relationships>` permettant de spécifier les opérateurs entre les tâches. Dans la partie `<context>`, le concepteur peut spécifier un contexte dans lequel un utilisateur ayant une autorisation de supervision de niveau « 3 » est défini. Selon le niveau d'autorisation, un filtrage des paramètres peut être effectué. Suite à ce filtrage, l'opérateur humain concerné pourra modifier des paramètres, ou encore les valeurs qu'il peut donner à un paramètre, selon son profil. Ensuite, le concepteur définit les ressources textuelles de chaque *composant abstrait* ou *conteneur abstrait* par rapport au contexte prédéfini. La Figure 8 illustre le modèle AMXML de la tâche *Recevoir alerte*.

```

<amxml>
<taskModel> <!--Définition des tâches du système-->
<task id="task1" name="Recevoir alerte" type="interactive">
<task id="task1.1" name="Afficher alerte" type="interactive">
<task id=" task1.1.1" name="Afficher données" type="interactive" function="showCtrlData"/>
<task id=" task1.1.2" name="Afficher message" type="interactive" function="showMsg"/>
</task>
<task id="task1.2" name="Régler paramètres" type="interactive">
<task id=" task1.2.1" name="Choisir paramètre" type="interactive" function="choice"/>
<task id=" task1.2.2" name="Choisir réaction" type="interactive" function="choice"/>
</task>
</taskModel>
<relationships> <!--Définition des liens entre les tâches-->
<link id="link1" type="concurrence" source="task1.1" target="task1.2"/>
<link id="link2" type="concurrence" source="task1.1.1" target="task1.1.2"/>
<link id="link3" type="enablingData" source="task1.2.1" target="task1.2.2"/>
</relationships>
</taskModel>
<auiModel> <!--Définition des conteneurs et les composants abstraits de l'IHM-->
<abstractContainer id="cnt1" name="Recevoir alerte">
<abstractContainer id="cnt1.1" name="Afficher alerte">
<abstractComponent id="cmp1.1.1" name="Afficher données"/>
<abstractComponent id="cmp1.1.2" name="Afficher message"/>
</abstractContainer>
<abstractContainer id="cnt1.2" name="Régler paramètres">
<abstractComponent id="cmp1.2.1" name="Choisir paramètre"/>
<abstractComponent id="cmp1.2.2" name="Choisir réaction"/>
</abstractContainer>
</abstractContainer>
</auiModel>
<contextModel> <!--Définition des contextes prévus-->
<context id="cntxt1">
<environment workstate="supervision"/><platform type="PC"/><user language="FR" permission="3"/>
</context>
</contextModel>
<resourceModel> <!--Définition des ressources des composants de l'IHM-->
<resource id="res1.2.1" component="cmp1.2.1" context="cntxt1" text="Choisir un paramètre"/>
<resource id="res1.2.2" component="cmp1.2.2" context="cntxt1" text="Régler le paramètre"/>
</resourceModel>
</amxml>

```

Figure 8. AMXML de l'IHM « Recevoir alerte »

### 3.3.2 Construction de la structure métier

Au niveau concret, la structure métier (cf. Figure 9) est construite (cf. §2.2.2) en appliquant des règles de construction (disponibles dans (Hariri, 2008)). Après avoir sélectionné les composants métier pour le système de supervision, les composants sélectionnés doivent être reliés en respectant la partie <relationships> du modèle AMXML, et à l'aide de patrons.

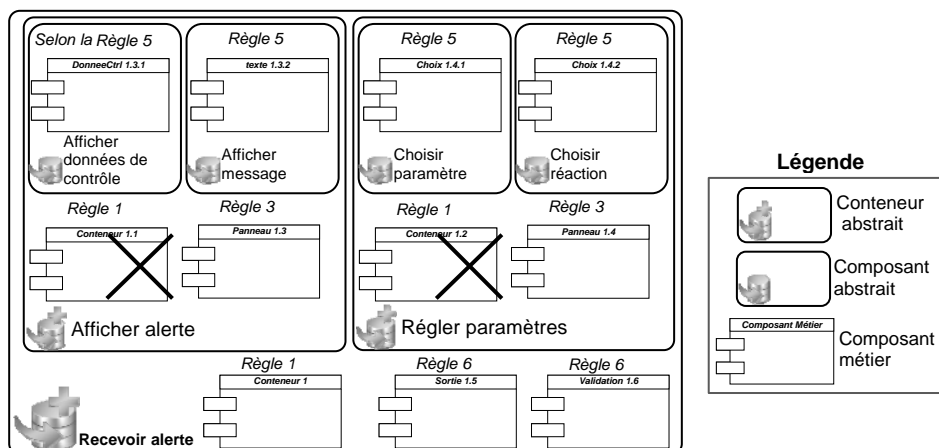


Figure 9. Génération de la structure métier

### 3.3.3 Adaptation de l'IHM primaire

Nous passons à la deuxième phase du niveau concret, où, une première version de l'IHM plastique est générée à base des composants de présentation (Figure 10). Tout d'abord, un composant de présentation (CP) de type fenêtre est choisi pour le composant métier "conteneur" 1. Deux CP de type "panneau" spécifiques aux deux composants métier "panneau" 1.3 et 1.4 sont sélectionnés. Ceux-ci sont ajoutés à la fenêtre du composant métier 1, et contiendront par la suite les CP sélectionnés pour les

composants métier 1.3.1, 1.3.2, 1.4.1 et 1.4.2. Le composant métier "DonneeCtrl" 1.3.1 choisit un CP de type "Cadran"<sup>2</sup>. Ensuite le composant métier "texte" 1.3.2 choisit une boîte de texte dans laquelle le message reçu peut être affiché. Comme l'espace d'affichage envisagé est suffisant, les composants métier "choix" 1.4.1 et 1.4.2 choisissent des CP de type "ListBox". Ensuite, les composants métier "sortie" et "validation" choisissent deux composants de type bouton, un pour quitter le système et l'autre pour valider le choix et/ou la saisie de l'utilisateur.

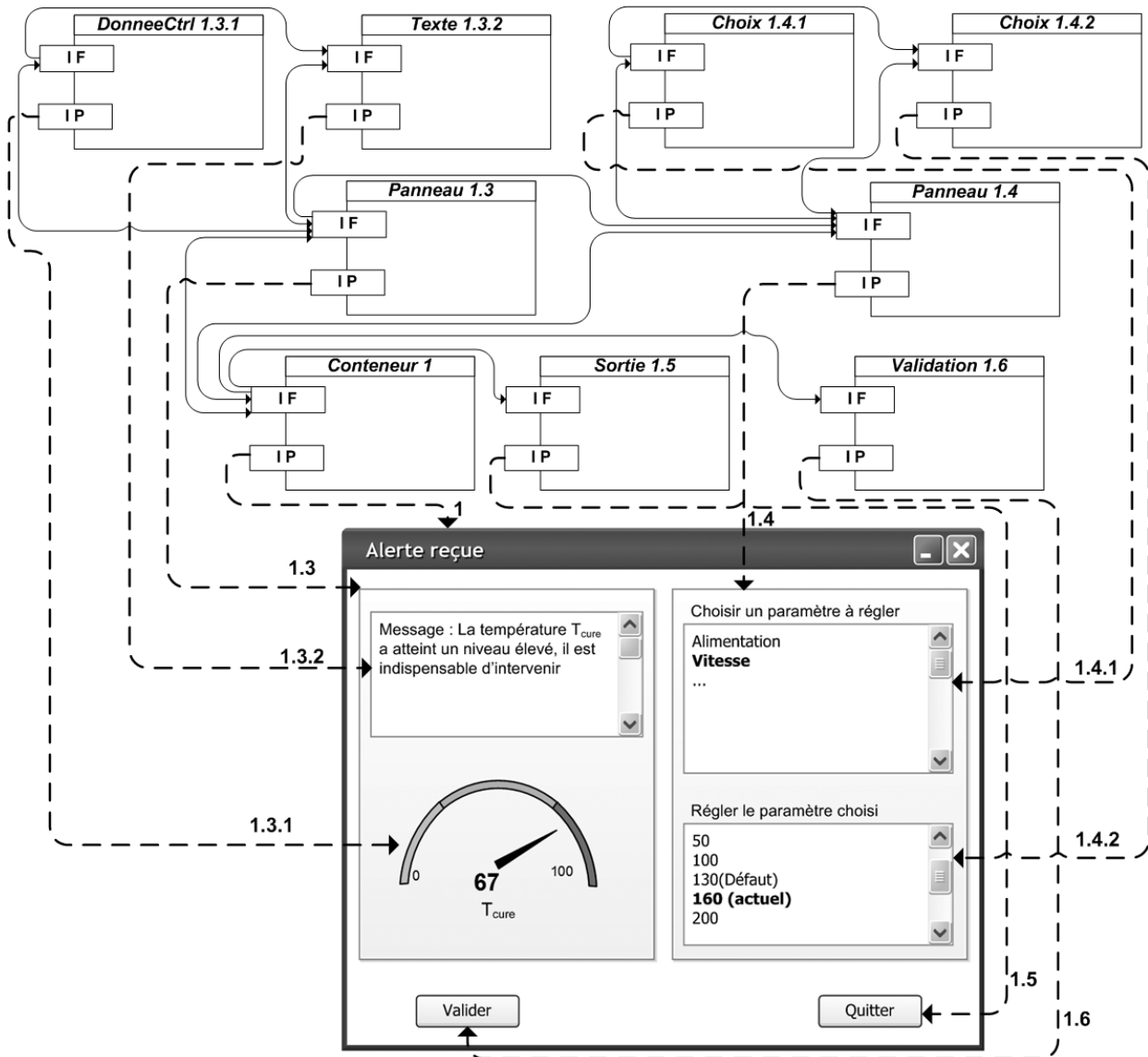


Figure 10. Première version du SI « Recevoir alerte »

### 3.3.4 Utilisation du SI plastique

Après la distribution du système de supervision sur le dispositif cible, l'utilisateur final commence à l'utiliser. Suite à une hausse de température  $T_{cuve}$  (rappelons que la température de la cuve  $T_{cuve}$  ne doit pas dépasser les 55°C, voir §3.1), l'IHM générée dans la précédente partie s'affiche automatiquement à l'utilisateur (de profil : opérateur de supervision dans la salle de contrôle). Ce dernier commence à interagir avec le SI, suite à l'apparition de l'alerte à traiter en urgence. En même temps, grâce à des équipements de capture du contexte, le contexte d'usage dans le SI est actualisé avec des nouvelles informations contextuelles capturées. Le processus d'apprentissage peut déduire des réactions répétées de l'utilisateur suite à des éventuels changements contextuels et de les ajouter à la base de connaissance.

### 3.3.5 Adaptation vivante

L'opérateur qui a reçu le message d'alerte, quitte la salle pour vérifier, sur place, l'état de la machine. Pour cela, il transfère son travail sur un PDA. Ce dernier est connecté au réseau. Ensuite le terminal de l'utilisateur détecte le PDA, et l'utilisateur fait migrer l'IHM vers la nouvelle plate-forme. Suite à l'important changement au niveau du contexte lié à la plate-forme, l'adaptation vivante appelle les composants métier à changer leur facette de présentation en adéquation avec les nouvelles caractéristiques de la plate-forme (PDA, possédant un espace d'affichage limité de 360x240). Les composants métier "Choix" 1.4.1 et 1.4.2, remplacent les composants de présentation "ListBox" par ceux de type "ComboBox" (moins d'espace d'affichage requis). Enfin, le

<sup>2</sup> Un composant de présentation de type "Cadran" permet d'afficher une valeur dans un instrument d'indication. Il est équipé d'un pointeur qui indique la valeur avec une variété d'informations qui indiquent par exemple le niveau normal et celui d'urgence d'une valeur.

composant métier "DonneeCtrl" choisit un composant de présentation de type "Barre". Celui-ci nécessite un petit espace d'affichage pour une valeur de la donnée de contrôle. Ensuite le patron "arrangeur" effectue l'arrangement des composants de présentation.

On suppose que l'utilisateur n'a pas pu intervenir à cause d'un manque de connaissance profonde relativement au problème qui est inhabituel. C'est pourquoi l'alerte est transférée à un opérateur d'astreinte (expert du domaine). Ce dernier est équipé d'un téléphone portable possédant un écran d'affichage 320x240. Le processus d'adaptation vivante est relancé. Le composant métier "DonneeCtrl" choisit de nouveau un composant de présentation de type "Texte". Celui-ci nécessite un très petit espace d'affichage. Ensuite les deux cadres sur l'IHM correspondant aux facettes de présentation des composants métier "panneau" 1.3 et 1.4 sont enlevés (en respectant le guide de style du téléphone portable). La liste des paramètres et des réactions est mis à jour suite au changement du profil de l'utilisateur (en particulier le niveau d'autorisations) (cf. Figure 11, partie droite). Enfin, le patron "Régleur de taille et de localisation" (cf. §2.1.3) recalcule la taille et la localisation considérées comme optimales pour les composants de présentation.



Figure 11. IHM adaptée pour un PDA, régénération pour un téléphone portable

#### 4. Conclusion et perspectives

Les travaux présentés dans cet article visent à proposer une démarche de génération d'IHM adaptable au contexte d'interaction pour la prise en compte de l'utilisateur dans le SI. Nous avons proposé une méthode en quatre étapes servant le processus de conception d'IHM plastique, cette méthode s'appuyant sur cinq principes : (1) le principe de conception d'IHM adaptable au contexte d'interaction de l'utilisateur, favorise la capitalisation et la réutilisation de connaissances de conception ; (2) la réification (traduction) des spécifications abstraites de SI d'autres méthodes classiques, orientées multicibles, ou orientées plasticité dans notre démarche, permet de faire le lien avec ces autres méthodes ; (3) l'utilisation des composants métier qui favorise la capitalisation et la réutilisation des composants dans l'architecture du SI plastique, de même que la minimisation des efforts de développement et de maintenance ; (4) l'architecture proposée pour les SI plastiques fait un découpage clair entre les différentes couches du système ; (5) le principe de choix et changement dynamique des composants de présentation qui favorisent l'adaptation de l'IHM au contexte d'interaction de l'utilisateur.

Il existe de nombreuses perspectives de recherche à ce travail. Il s'agirait de laisser le choix entre l'embarquement des composants métier correspondant à une IHM ou le téléchargement lors de l'exécution. L'intégration des composants de type *Wcomp* (Cheung *et al.*, 2003) dans notre démarche pourrait être étudiée. Les composants *Wcomp* sont de type mixte ; autrement dit un composant logiciel représente un équipement matériel. Ces composants nous permettraient d'effectuer des adaptations à l'interface physique (environnement de travail). Les patrons peuvent actuellement proposer plusieurs solutions pour un problème donné dans un contexte donné, il faudrait alors affiner le raisonnement lié à l'utilisation des patrons. Les patrons qui ont été proposés sont définis textuellement dans des fichiers XML. Un formalisme tel que P-Sigma (Conte *et al.* 2001) associé à la définition d'une ontologie pourrait leur permettre d'être directement exploitables. Enfin, à moyen terme, la démarche proposée pourra être enrichie de mécanismes et outils spécifiques ou existants (UsiXML) pour supporter les différentes activités.

#### Remerciements

Cette recherche a été financée partiellement par la région Nord-Pas de Calais, le Ministère de l'Education Nationale, de la Recherche et de la Technologie, le FEDER et l'université de Concordia-Canada. Les auteurs les remercient, de même que les relecteurs anonymes pour leurs remarques pertinentes.

#### 5. Bibliographie

- Alexander C., Ishikawa S., Silverstein M., Fiskdahl-King I., Angel S., *A pattern language*. Oxford University Press, New York, 1977.
- Bandelloni R. et Paternò F., « Migratory user interfaces able to adapt to various interaction platforms », *Int. J. Human-Computer Studies*, Vol. 60, n° 5-6, 2004, p. 621-639.
- Barbier F., Cauvet C., Oussalah M., Rieu D., Bennisri S., Souveyet C., « Composants dans l'ingénierie des systèmes d'information : concepts clés et techniques de réutilisation ». Actes des 2e Assises nationales du GDRI3 - Information Interaction Intelligence, Nancy, Cepaduès, 2002, p. 95-117.

- Brown P., Bovey J., Chen X., Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Communications*, Vol. 2, n° 1, 1997, p. 1–9.
- Bayle, E., Putting it All Together: Towards a Pattern Language for Interaction Design, *SIGCHI Bulletin*, Vol 30, n° 1, 1998, p.17-24.
- Calvary G., Coutaz J., « Plasticité des Interfaces: une nécessité! », Actes des deuxièmes Assises nationales du GDR I3, Cépaduès Editions, J. Le Maître (Ed), Nancy, Décembre 2002a, p. 247-261.
- Calvary G., Coutaz J., Thevenin D., Bouillon L., Florins M., Limbourg Q., Souchon N., Vanderdonckt J., Marucci L., Paternò F., Santoro C., The CAMELEON Reference Framework, Deliverable D1.1, 2002b.
- Calvary G., Coutaz J., Thevenin D., Limbourg Q., Bouillon L., Vanderdonckt J., « A unifying reference framework for multi-target user interface ». *Interacting with Computers*, Vol. 15, n° 3, 2003, p. 289-308.
- Cauvet C., Semmak F., La réutilisation dans l'ingénierie des systèmes d'information. *Génie objet – Analyse et conception de l'évolution*, Oussalah C. (ed.), Hermès Science Publications, Paris, 1999, p. 25-55.
- Cheung D., Fuchet J., Grillon F., Joulie G. et Tigli J.Y., « Wcomp : Rapid Application Development Toolkit for Wearable computer based on Java », In Proceedings of the IEEE Int. Conference on Systems, Man and Cybernetics, Vol. 5, 2003, p. 4198–4203.
- Conte A., Giraudin J.P., Hassine I., Rieu D., « Un environnement et un formalisme pour la définition, la gestion et l'application de patrons », *Revue ISI Ingénierie des Systèmes d'Information*, Vol. 6, n° 2, Hermès, 2001.
- Demeure A., Calvary G., Coutaz J., Vanderdonckt J., « The Comet Inspector: Manipulating Multiple User Interface Representations Simultaneously », *Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006* (Bucharest, 6-8 June 2006), Chapter 13, Springer-Verlag, Berlin, 2006, p. 167-174.
- Dey A.K., Providing Architectural Support for Building Context-Aware Applications, PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.
- Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns, Catalogue de modèles de conception réutilisables*. Vuivert, Paris, 1994.
- Garlan D., Cheng S.W., Huang A.C., Schmerl B., Steenkiste P., « Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure », *Computer*, 37 (10), 2004, p. 46-54.
- Gellner M., Forbrig P., « Modeling the Usability Evaluation Process with the Perspective of Developing a Computer Aided Usability Evaluation (CAUE) System ». *Proc. of INTERACT 2001 Workshop on "Usability throughout the entire systems development lifecycle"*, Tokyo, Japan, July 10-14 2001.
- Grolaux D., Van Roy P., Vanderdonckt J., « QtK: A Mixed Model-Based Approach to Designing Executable User Interfaces », *Proc. of 8th IFIP Working Conference on Engineering for Human-Computer Interaction EHCI01*, Toronto, 11-13 mai 2001.
- Hariri A., Tabary D., Kolski C., « Plastic HCI generation from its abstract model ». *Proc. of IASTED-HCI Int. Conf. on Human-Computer Interaction* (November 14-16, 2005, Phoenix, USA), ACTA Press, Anaheim, USA, 2005, p. 246-251.
- Hariri A., Lepreux S., Tabary D., Kolski C., « Principe d'adaptation d'interfaces home-machine dans les systèmes d'information en fonction du contexte et de l'utilisateur », Actes d'INFOSID 2007, atelier PECUSI, (mai 2007, Perros-Guirec, France)
- Hariri A., Tabary D., Lepreux S., Kolski C., « Context aware Business adaptation toward User Interface adaptation » *Communications of SIWN*, Vol. 3, 2008, p. 46-52.
- Hariri A., « Contribution à une méthode de conception et génération d'interface homme-machine plastique », Thèse de doctorat, Université de Valenciennes, juin, 2008.
- Hassine I., Rieu D., Bousnaas F., Seghrouchni O., « Symphony : un modèle conceptuel de composants métier ». *Connaissances métier en ingénierie des systèmes d'information*, *Ingénierie des systèmes d'information*, Cauvet, C. (ed.). Hermès, Paris, 2002, p. 35-59.
- Henricksen K., Indulska J., « a Software Engineering Framework for Context-Aware Pervasive Computing », *Proc. International Conf. Pervasive Computing and communications*, Florida, USA, 2004.
- Lepreux S., Vanderdonckt J., Michotte B., "Visual Design of User Interfaces by (De)composition", In Proceedings of 13th Int. Workshop on Design, Specification and Verification of Interactive Systems DSV-IS'2006 (Dublin, July 2006), Doherty G. and Blandford A. (Eds.), Springer, LNCS, Berlin, 2006, p. 157–170.
- Limbourg Q., Vanderdonckt J., Michotte B., Bouillon L., Lopez V., « UsiXML: a Language Supporting Multi-Path Development of User Interfaces ». In *Proc. of 9th IFIP Working Conf. on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004* (Hamburg, July 11-13, 2004). LNCS 3425, Springer-Verlag, Berlin, 2005, p. 200–220.
- Moussa F., Kolski C., Riahi M., « Analyse des dysfonctionnements des systèmes complexes en amont de la conception des IHM : apports, difficultés, et étude de cas », *Revue d'Interaction Homme-Machine (RIHM)*, Vol. 7, 2006, p. 79–111.
- Nielsen J., *Usability engineering*, Boston, Academic Press, 1993.
- Paterno F., *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, Londres, 2000.
- Paternò F., Santoro C., « One Model, Many Interfaces », *Computer Aided-Design of User Interfaces*, Kolski C., Vanderdonckt J. (Eds.), kluwer Academic pub., 2002, p. 143-154.
- Petersen J., May M. « Scale transformations and information presentation in supervisory control ». *Int. J. Human-Computer Studies*, Vol. 64, n° 5, 2006, p. 405–419.
- Riahi M. « Contribution à l'élaboration d'une méthodologie de spécification, de vérification et de génération semi-automatique d'interfaces homme-machine : Application à l'outil Ergo-Conceptor+ », Thèse de doctorat, Université de Valenciennes, 2004.
- Schilit B., Theimer M., « Disseminating active map information to mobile hosts ». *IEEE Network*, Vol. 8, n° 5, 1994, p. 22–32.
- Tabary D. « Contribution à TOOD, une méthode à base de modèles pour la spécification et la conception des systèmes interactifs ». Thèse de doctorat, Univ.Valenciennes, déc.2001.
- Thevenin D., Coutaz J., « Plasticity of User Interfaces: Framework and Research Agenda ». *Proc. of 7th IFIP Int. Conf. on HCI Interact'99* (1999, Edinburgh, Scotland), p. 110-117.