



**Master Recherche Informatique
Réseaux, Télécommunications et Services
Systèmes Coopératifs**

Thème : Plasticité

Articles:

- ♦ CAMELEON-RT : a Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces

Auteurs: Lionel Balme, Alexandre Demeure, Nicolas Barralon, Joëlle Coutaz, Gaëlle Calvary

- ♦ A Reference Framework for the Development of Plastic User Interfaces

Auteurs: David Thevenin, Joëlle Coutaz, Gaëlle Calvary

Présentés par :

- ♦ Betânia STEFFEN ABDALLAH GONÇALVES
- ♦ Sérgio RODRIGUES DE MORAIS

SOMMAIRE

1. INTRODUCTION	4
2. RESUME DES ARTICLES	6
CAMELEON-RT : A SOFTWARE ARCHITECTURE REFERENCE MODEL FOR DISTRIBUTED, MIGRATABLE, AND PLASTIC USER INTERFACE	
	6
1. <i>CamNote</i>	7
2. <i>The Interaction Abstract Machine (I-AM)</i>	7
3. <i>The CAMELEON-RT Architecture Reference Model</i>	8
4. <i>Critique</i>	9
A REFERENCE FRAMEWORK FOR THE DEVELOPMENT OF PLASTIC USER INTERFACES	
	10
1. <i>Introduction</i>	10
2. <i>Le Plastic UI Snowflake</i>	10
3. <i>Le "Process Reference Framework" pour les interfaces multi cibles et plastiques</i>	12
4. <i>ARTstudio : application du "Process Reference Framework"</i>	15
5. <i>Critique</i>	17
3. CONCLUSION	19
4. CARTE DE CONCEPTS	20
5. BIBLIOGRAPHIE	22

1. INTRODUCTION

La grande variété des dispositifs informatiques (téléphones mobiles, PDAs, WEBTV, etc.) présents dans le marché aujourd'hui, font apparaître des nouveaux défis dans la conception d'applicatifs que soient compatibles avec tous ces différents dispositifs. En plus, nos objets quotidiens s'augmentent d'électronique, ce qu'il fait qu'ils deviennent des objets possibles d'interaction.

Pour citer quelque de ces défis, on a la construction et maintenance des versions de interfaces utilisateurs parmi les dispositifs, la vérification de la consistance des versions pour assurer l'interaction entre les divers dispositifs et la conception de l'habilité de s'adapter dynamiquement à des changements d'environnement.

Dans ce contexte de créer des interfaces qui sont adaptables à plusieurs types de dispositifs différents, qu'on introduit le concept de plasticité. Le terme *plasticité* a été inspiré de la plasticité des matériaux qui s'étirent et se contractent sous des contraintes naturels sans rompre et en préservant leur utilisabilité. Alors, la plasticité d'une interface dénote sa capacité de s'adapter au contexte d'usage en préservant son utilisabilité.

Il existe encore quelques concepts qui sont important d'être définis lorsqu'on parle de plasticité d'interfaces :

- ♦ Espace Interactive : L'espace interactive est la combinaison de trois choses complémentaires qui sont le lieu physique où l'interaction se passe, les ressources informatiques, de réseaux et d'interaction disponibles et le monde digital qui supporte des activités humaines.
- ♦ Distribution d'interfaces : Une interface est distribuée lorsque elle utilise des ressources qui sont distribués parmi un cluster.
- ♦ Migration d'interfaces : La migration d'interfaces correspond au transfert du total ou d'une partie de l'interface utilisateur parmi des différentes ressources d'interaction.
- ♦ Contexte d'utilisation et cible : Le contexte d'utilisation est la situation courant d'utilisation du système lors de son exécution. La cible est la situation d'utilisation envisagée par le concepteur de l'interface. Ils sont formés par la triple utilisateur, plate-forme et environnement.
- ♦ Interface Multi cible et Interface plastique : Une interface multi cible est une interface capable de s'adapter à plusieurs cibles différentes. Par contre, une interface plastique est une interface multi cible que préserve l'utilisabilité dans les différentes cibles.

Alors, après avoir défini quelques concepts importants fortement liés à la plasticité, il est possible de comprendre les deux *frameworks* présentés dans les articles traités : le "CAMELEON-RT" et le "Process Reference Framework"

2. RESUME DES ARTICLES

CAMELEON-RT : A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interface

Aujourd'hui, une nouvelle évolution se prépare avec la convergence des réseaux ad hoc sans fil, de micro et nano systèmes et la maturation de trente années de recherche en systèmes répartis, en perception artificielle et en interaction homme machine (IHM). Cette convergence de technologies permet d'envisager le passage d'une informatique confinée et statique, à une informatique ambiante favorisant l'émergence opportuniste d'écosystèmes.

Dans cette vision, les espaces interactifs prennent la forme d'îlots autonomes. Ces derniers sont sans cesse en mutation puisque l'utilisateur peut dynamiquement faire apparaître, composer, prêter ou emprunter les ressources d'interaction disponibles dans son espace.

La métaphore du bureau supposant un utilisateur devant un ordinateur composé d'un clavier, d'un écran et d'une souris n'est plus suffisante. Les IHM conventionnelles WIMP (Windows, Icons, Menus, Pointing) supposent l'existence de dispositifs d'interaction dont le nombre et le type sont connus par avance. Avec la réalité mixte, tout objet familier augmenté est un dispositif d'interaction donnant lieu à une grande diversité de types. Avec l'informatique ambiante, à la diversité des types s'ajoute la présence incertaine de dispositifs qui peuvent être empruntés, prêtés, composés et découverts dynamiquement. C'est à dire que il faut rendre compte des problèmes de distribution et de migration.

Ces deux dernier points, imposent, pour garantir l'utilisabilité des IHM que celles-ci soient plastiques afin de s'adapter aux changement de l'espace interactif. CAMELEON-RT est un modèle de référence de architecture utilisé pour faire des comparaisons et analyses sur des outils déjà existants et aussi pour développer de nouvelles infrastructures pour les interfaces distribuées, migrantes et plastiques. Ce modèle de architecture proposée par CAMELEON-RT intégré tous les éléments nécessaires pour supporter distribution, migration et plasticité des interfaces d'une façon dynamique en clusters hétérogènes et dynamiques.

Pour exemplifier ça, deux études de cas sont présentées.

1. CamNote

CamNote (de CAMELEON Note) est un projecteur de *slides* qui tourne sur une plate-forme hétérogène et dynamique. Cette plate-forme fait le changement allant d'un simple PC à un cluster composé d'un PC plus un PDA. À principe le PDA s'occupe de fonction de contrôle, mais une fois que le PDA disparaît du cluster, le panneau de contrôle migre automatiquement vers l'écran du PC. Si le PDA réapparaît, l'interface change automatiquement à nouveaux vers la configuration de l'image 1.

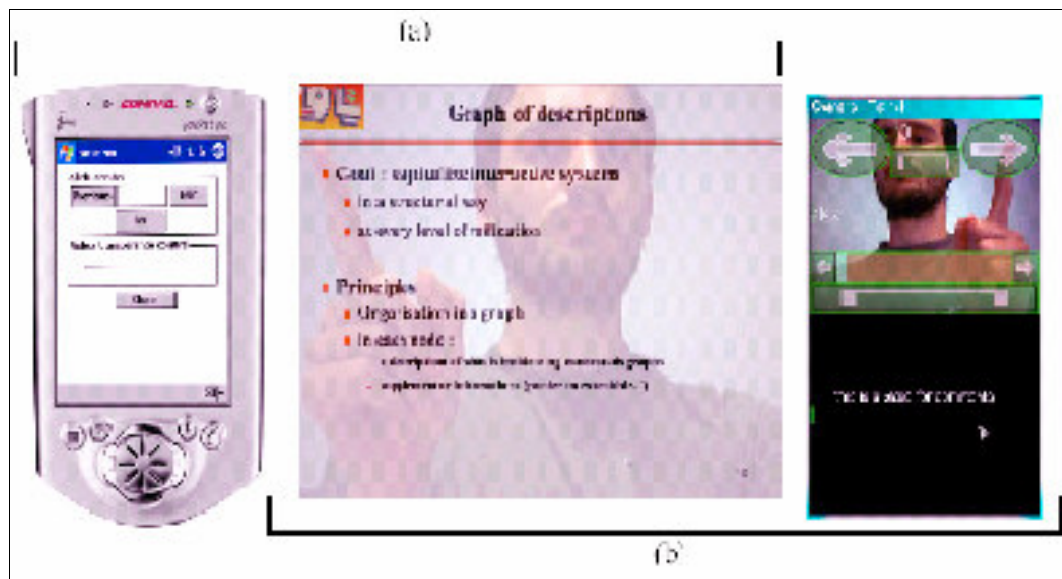


Image 1 : Interface Distribuée

2. The Interaction Abstract Machine (I-AM)

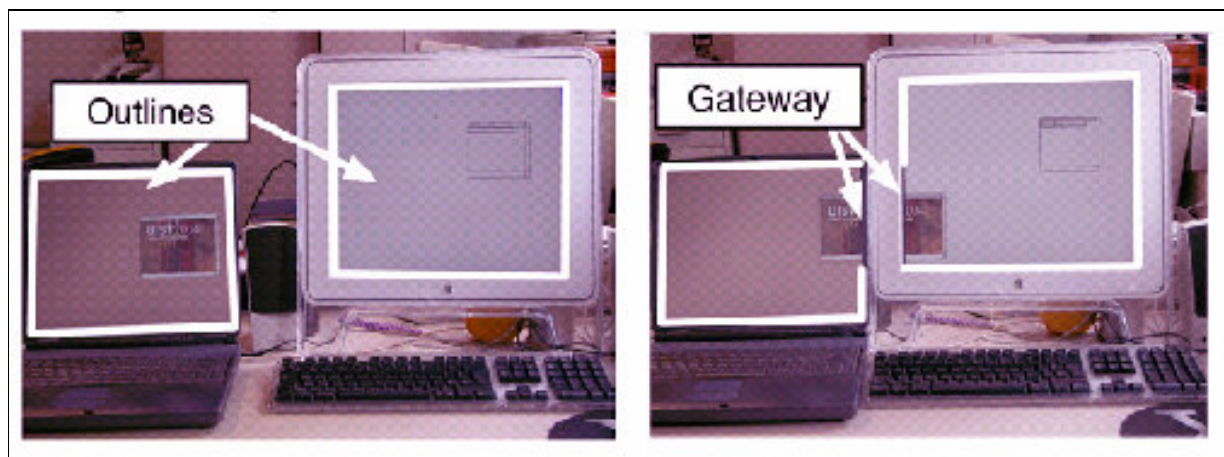


Image 2: Migration d'interfaces

L'I-AM supporte une configuration dynamique de ressources d'interaction et forme un unique espace logique d'interaction. Ces ressources sont administrées par workstations qui tournent des systèmes d'exploration différents (MacOS X, Windows NT, XP). Les utilisateurs peuvent distribuer et faire migrer des interfaces au niveau pixel comme si ces interfaces étaient traitées pour un seul PC. Avec un espace interactif n'importe quel instrument peut être utilisé pour modifier n'importe quel interacteur.

3. The CAMELEON-RT Architecture Reference Model

CAMELEON-RT est structuré en trois niveaux d'abstraction: la couche des systèmes interactives, la couche des plate-formes et le coeur de la architecture, le DMP-Middleware.

La couche des plate-formes est formée par le hardware et les systèmes d'exploitation lesquels forment la base d'un espace d'interaction.

La couche des systèmes interactive inclut les systèmes interactifs (CamNote, par exemple) qui sont tournés pour les utilisateurs dans un espace interactive. Les Interfaces Meta Utilisateur sont pour les espaces interactives le même que le *desktop* est pour les stations de travail conventionnelles.

La couche DMP-Middleware a pour but satisfaire trois classes de réquisition: modeler le espace physique, supporter clusters hétérogènes et dynamiques, et adaptation des UI quand ce succède distribution et migration. Pour chaque un de ces réquisition correspond un service de la couche DMP: "*contexte infrastructure*", "*plateforme manager*" avec son "*interaction toolkit*", et "*open-adaptation manager*".

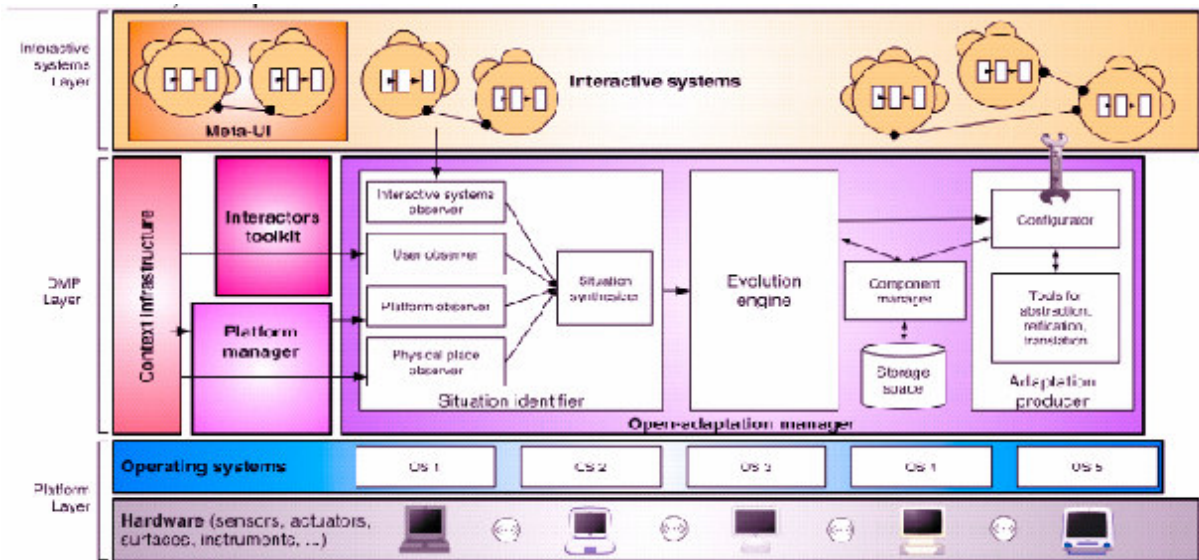


Image 3 : CAMELEON framework

4. Critique

CAMELEON-RT est une décomposition fonctionnelle qui couvre tous les aspects des DMP - UI. Il n'est pas une architecture de implémentation et il laisse libre le choix de styles de architectures. Cependant CAMELEON-RT doit encore être raffiné et évalué avec des expérimentations futures.

A Reference Framework for the Development of Plastic User Interfaces

1. Introduction

L'adaptation des interfaces utilisateurs est un vrai défi. Cet article propose un *framework* que clarifie la nature de l'adaptation des interfaces plastiques à partir de la perspective de la conception des *softwares*.

Deux composants complémentaires seront présentés. Le premier est un espace taxonomique appelé "*The Plastic UI Snowflake*" et le deuxième est le "*Process Framework*". Ils seront expliqués dans les sessions suivantes.

2. Le Plastic UI Snowflake

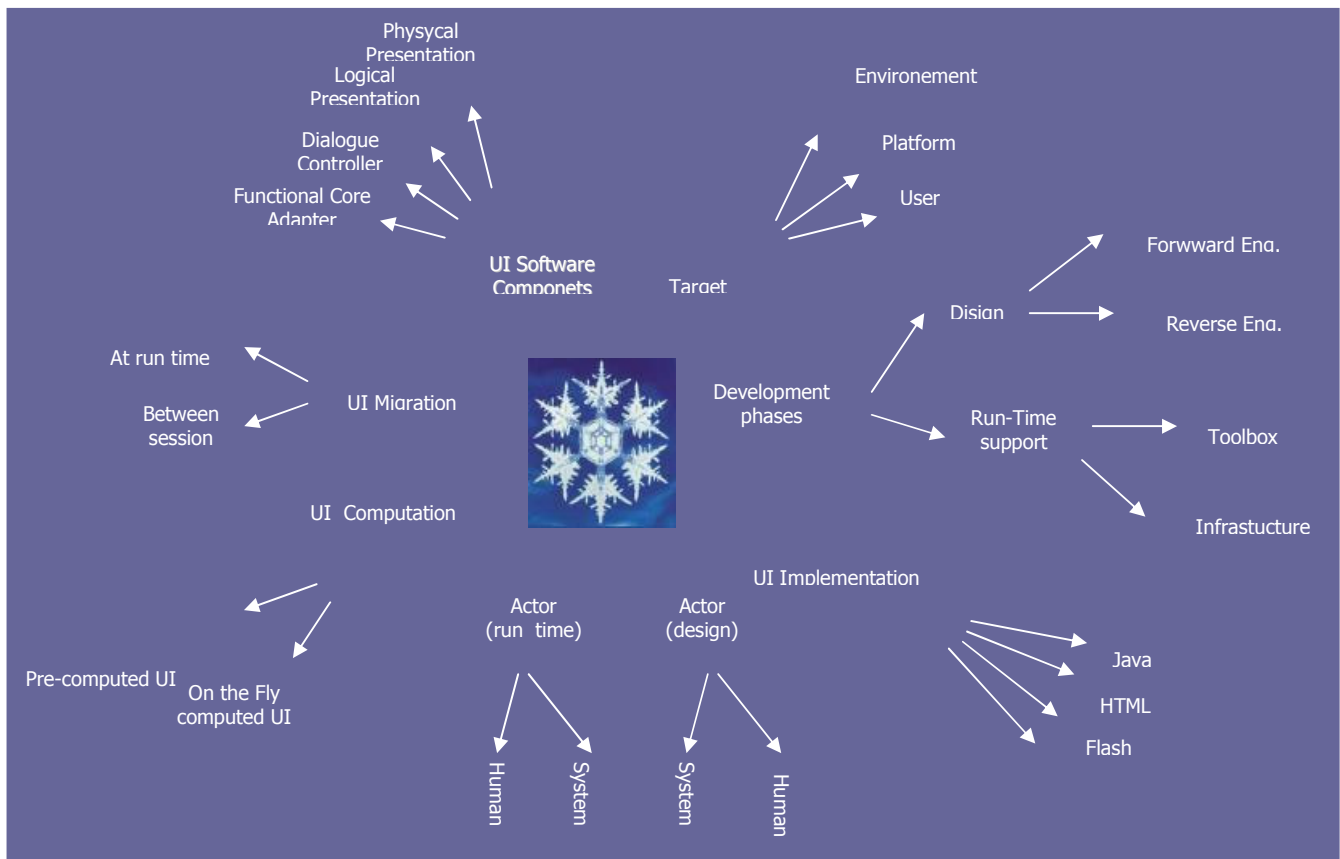


Image 4 : Le "*Plastic UI Snowflake*"

Le "*Plastic UI Snowflake*" peut être utilisé soit pour la caractérisation des outils qui existent déjà soit pour exprimer les contraintes pour des outils futures. Chaque branche montre des caractéristiques importantes pour la plasticité des interfaces.

En suite, chaque branche du "*Plastic UI Snowflake*" est détaillée.

- ♦ Sensibilité de la cible (*target*) :

Première caractéristique à se définir lorsque on parle des outils pour la plasticité. Il faut définir si l'outil sera soit multi plate-forme, soit multi environnement, etc.

Il n'y a pas d'outil connu qui supporte tous les trois dimensions de la plasticité.

- ♦ Classes d'outils (*Classes of software tools: Development phases and UI Implementation*):

Il faut distinguer les outils utilisés dans la phase de conception d'un système et les outils utilisés pour l'implémentation des interfaces et les mécanismes utilisés pendant l'exécution.

En ce qui concerne la phase d'implémentation (code), elle est basée sur l'infrastructure du *framework* et les outils. Il est important de bien choisir la plate-forme qui s'adapte le plus à la conception de l'interface concerné.

- ♦ Acteurs (*Actors*) :

Pendant la phase de conception, la plasticité peut être implémenté soit par le concepteur soit par les outils de la plat-forme utilisée pour la conception.

Pendant la phase d'exécution, l'interface peut s'adapter à une demande d'utilisateur et aussi elle peut s'adapter automatiquement.

- ♦ Calcul des interfaces multi cibles et plastiques (*UI Computation*) :

En ce qui concerne les interfaces pré calculés, l'adaptation est fait dans la phase de conception pour chaque cycle envisagé.

Les interfaces "*On the fly*" s'adaptent dynamiquement pendant l'exécution.

Il est possible de concevoir des interfaces hybrides, qui utilisent les deux façons de s'adapter aux différentes cibles.

- ♦ Les composants du *software* de l'interface (*UI Software Components*) :

Certains composants du *software* sont affectés pendant l'adaptation d'une interface plastique et multi cible. Ces composants sont : sa présentation physique, sa présentation logique, le contrôle de dialogue, l'adaptateur fonctionnel du *core*.

- ♦ Migration des Interfaces (UI Migration) :

La migration des interfaces correspond au transfert d'une interface entre des plate-formes différentes. Il est possible de faire la migration soit pendant la phase d'exécution soit entre des sessions.

L'article "CAMELEON-RT : a Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces" explique plus en détail le processus de migration des interfaces.

3. Le "Process Reference Framework" pour les interfaces multi cibles et plastiques.

Le "Process Reference Framework" fourni aux concepteurs les principes généraux pour la structuration et la compréhension du processus de conception d'interfaces multi cibles et plastiques.

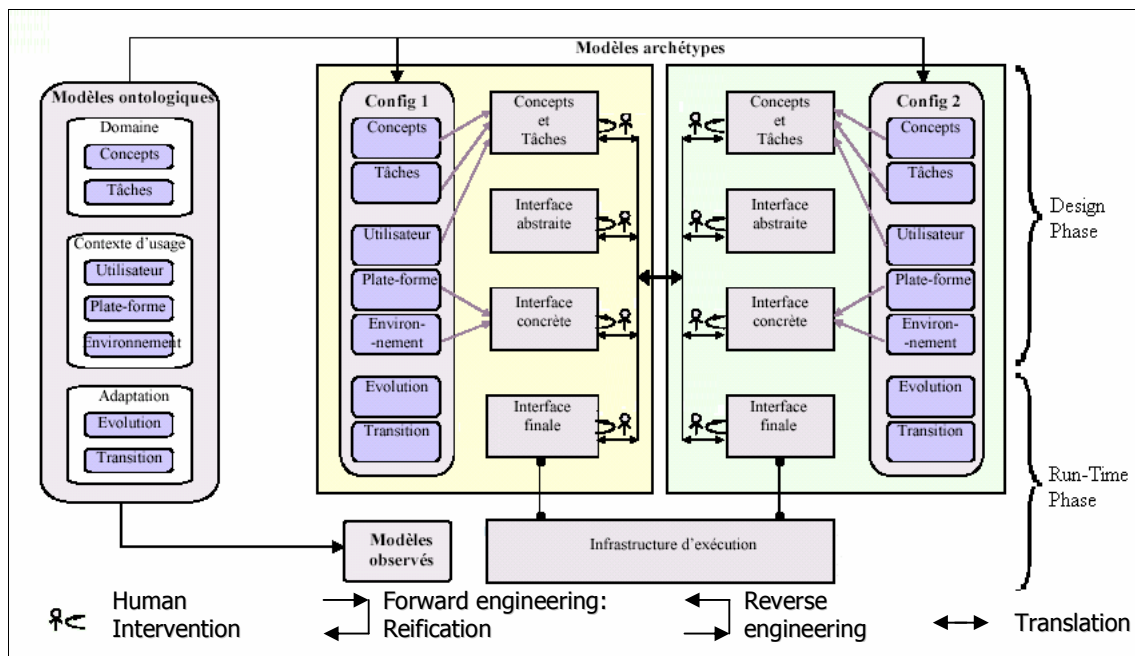


Image 5 : Le "Process Reference Framework"

Description Général

- ♦ Modèles et cycle de vie :

Les modèles sont de base pour la génération et l'abstraction de code et ils réduisent le coût de sa production. Chaque modèle correspond à un pas dans le cycle de vie.

Plasticité des Interfaces

- Modèles Ontologiques : ces sont des meta modèles qui définissent les dimensions clés de la plasticité.
 - Modèles Archétypes : ils sont des spécifications d'entrée pour la phase de conception d'un système interactive.
 - Modèles Observés : ces sont des modèles exécutables qui support l'adaptation.
- ♦ Couverture des Modèles :

Il y a 3 classes de modèles qui peuvent être ontologiques, archétypes ou observés:

- Modèles de Domaine : comprend les entités et les tâches que l'utilisateur manipule.
- Modèles de Contexte d'utilisation : une cible (utilisateur, plate-forme, environnement).
- Modèles d'adaptation : adaptation du système lorsque le contexte d'utilisation et/ou la cible change.

La phase de conception

Dans la phase de conception, le "*Process Reference Framework*" fourni les principes pour les processus de développement d'interfaces multi cibles et plastiques.

- ♦ Réification : il est le processus de passer d'une description de haut niveau à un code exécutable.



Image 6 : Le processus de réification à 4 pas

- ♦ Translation : transformation d'une description pour une cible dans une description pour une autre cible.
- ♦ Intervention Humaine : dans l'absence d'un outil, la réification et la translation peuvent être réalisés manuellement.
- ♦ Décoration : exceptions pour les standards ou les défauts.

- ♦ Factorisation : pour une classe, la factorisation produit une nouvelle description composée d'une description partagée par toutes les cibles et une description spécifique à chaque cible.

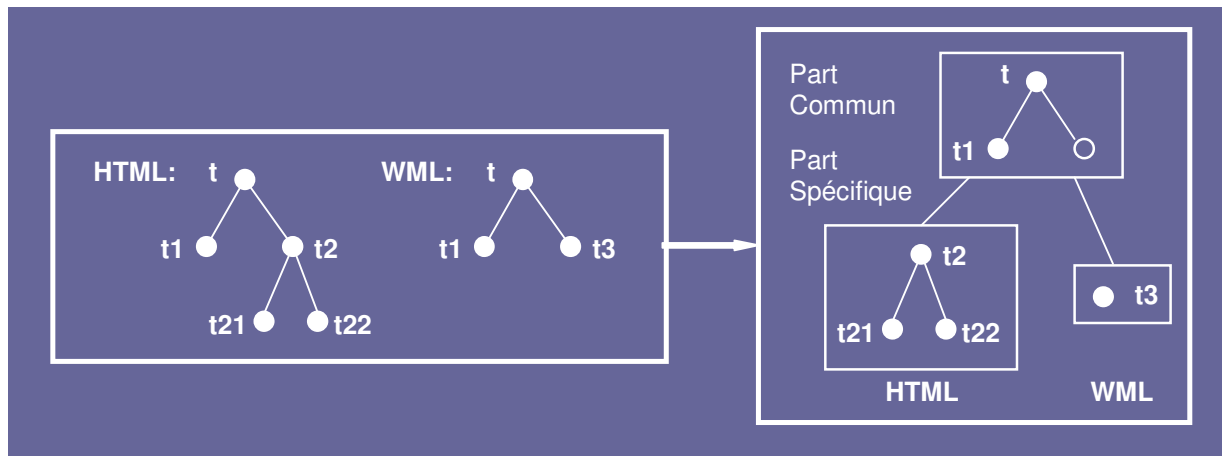


Image 7 : Exemple de factorisation

4. ARTstudio : application du "*Process Reference Framework*"

L'ARTstudio est un logiciel pour le développement d'interfaces pré calculé et multi plate-formes. Il utilise les principes du "*Process Reference Framework*".

Dans son implémentation courrent, il support le processus à 4-pas de réification, l'intervention humaine et la factorisation. Par contre, il ne support pas la translation et la décoration.

Pour mieux comprendre l'ARTstudio, on va utiliser un étude de cas : Le système de contrôle de chauffage d'une maison d'EDF. Ce système permet au utilisateur de contrôler la température de quelque pièces de sa maison dans des contexte d'utilisation différents, comme à la maison à partir d'un PDA, du bureau à partir d'un ordinateur de bureau connecté à l'Internet et d'un autre endroit quelconque à partir d'un téléphone portable (WAP). Les utilisateurs cibles sont des membres de famille adultes. Dans cet exemple, les interfaces doivent s'adapter aux différents dispositifs utilisés.

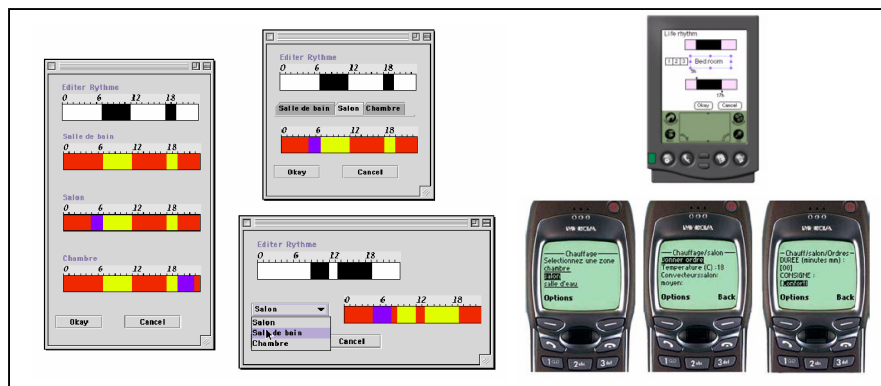


Image 8 : Le système de contrôle de chauffage d'une maison d'EDF

Après avoir créer le modèle de concepts de domaine (des noms pour les objets), on commence la modélisation de l'interface où chaque étape est un modèle.

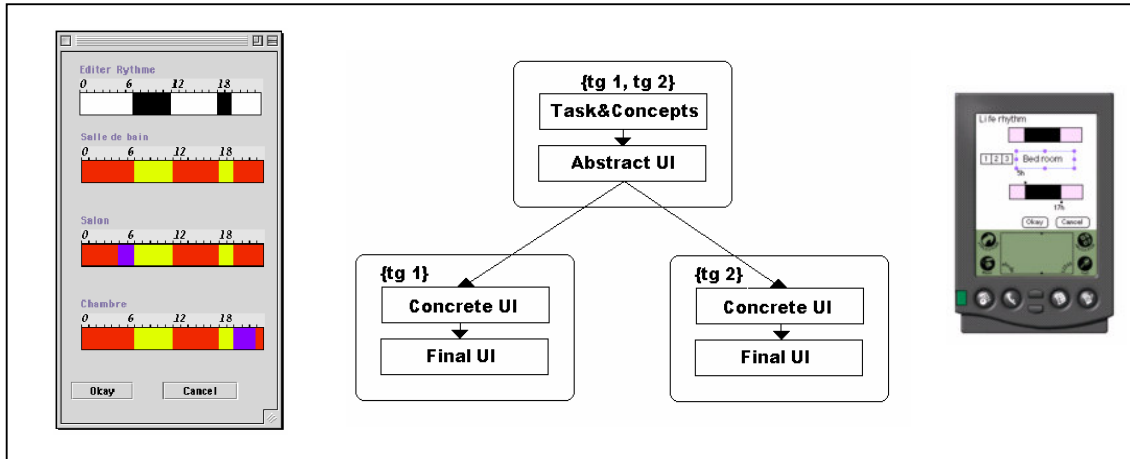


Image 9 : Instanciations du "Process Reference Framework" dans l'ARTstudio

♦ Le modèle de Tâches :

- Spécification du nom et du type de la tâche,
- Spécification d'un prologue et d'un épilogue (fonctions exécutées avant et après la tâche),
- Mettre en référence les concepts qui comprennent la tâche et les saisir par rapport à son niveau d'importance.

♦ Le modèle d'interface abstraite :

Une interface abstraite est modélisée comme un ensemble structuré de zones de travail où chaque zone de travail a une correspondance une à une avec le modèle de tâches. En plus, une zone de travail qui correspond à une tâche abstraite comprend les zones de travail qui correspondent aux sous tâches de la tâche abstraite.

♦ Le modèle d'interface concrète :

L'interface concrète est générée à partir de l'interface abstraite. Elle consiste d'un ensemble de fonctions de *mapping* :

- entre les zones de travail et les surfaces d'affichage,
- entre les concepts et les interacteurs,
- entre le schéma de navigation et les interacteurs de navigation.

L'interface finale est générée à partir de l'interface concrète et c'est l'interface finale qu'on voit à l'exécution.

5. Critique

L'article "A Reference Framework for the development of Plastic User Interfaces" montre bien la problématique de concevoir des interfaces plastiques e multi cibles.

Le *framework* présenté est très bien structuré et le "*Plastic UI Snowflake*" possibilite une structuration claire des différentes contraintes qui doivent être pris en compte dans la modélisation des interfaces.

Néanmoins, le "*Process Reference Framework*" n'a pas encore réussi à résoudre le problème de conception d'interfaces multi utilisateurs.

Le logiciel ARTstudio, que utilise le "*Process Reference Framework*", n'arrive pas à supporter la translation et la décoration, qui sont des processus importants lorsqu'on parle d'interfaces multi cibles et plastiques. En plus, le non support de la translation est incohérent avec son nom (**A**daptation through **R**eification and **T**ranslation **S**tudio).

3. CONCLUSION

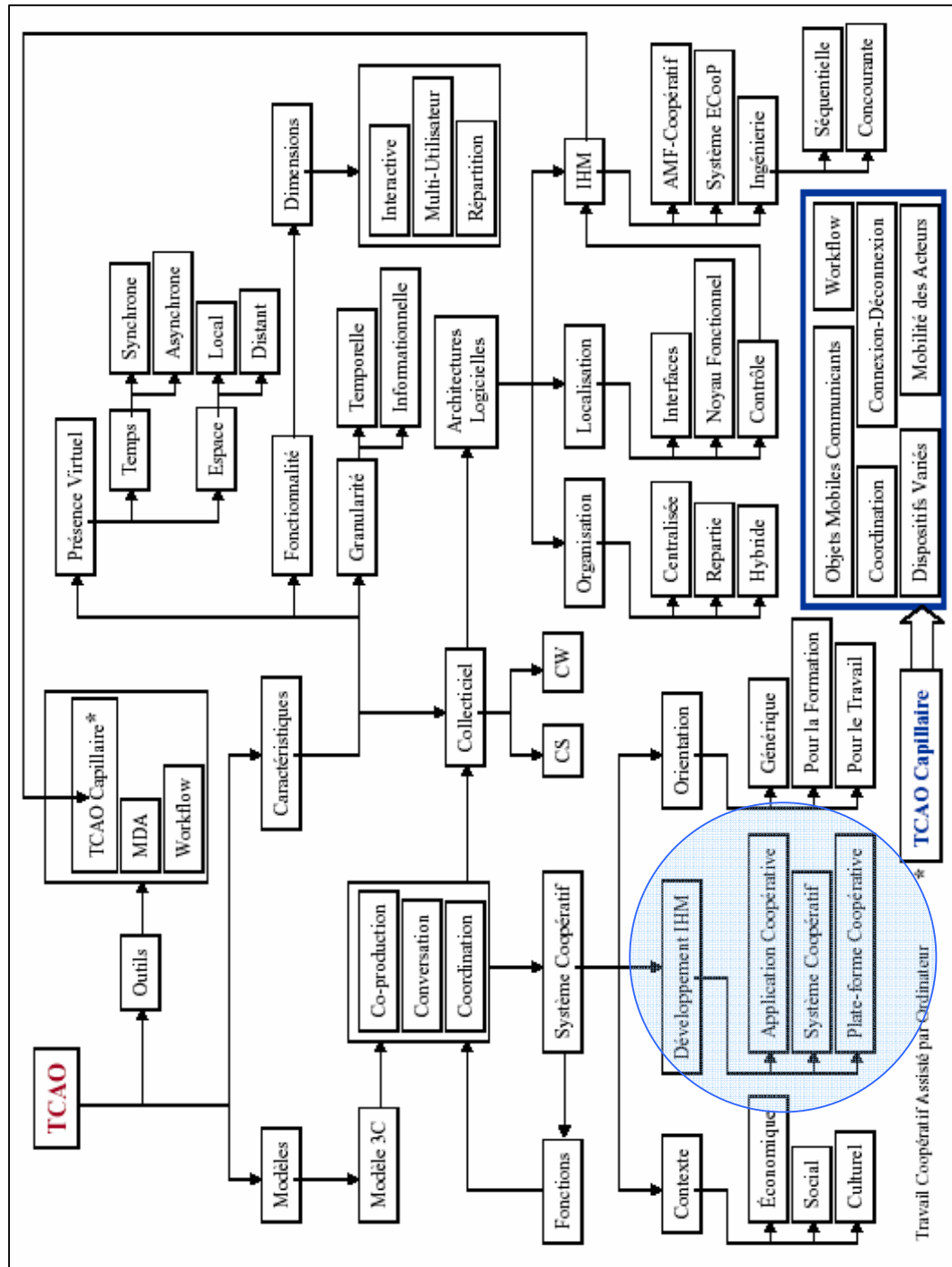
L'informatique ambiante présente de nouveaux challenges en termes d'interaction multi surface / multi instruments et de plasticité. Les services offerts par le bureau classique ne permettent plus de garantir les propriétés d'ergonomie habituellement appliquées. Il faut dorénavant prendre en charge : l'emprunt, le prêt et la composition de ressources d'interaction ; la migration, la distribution et par conséquence la plasticité des IHM avec ses services de négociation et de transition.

Les deux articles présentés traitent de cette problématique. Le premier, "CAMELEON-RT : a Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces", présente un modèle de architecture de référence qui peut être utilisé pour la comparaison et analyse des outils déjà existants où encore pour le développement de futures infrastructures pour UI distribuées, migrantes et plastiques. Le deuxième, "A Reference Framework for the Development of Plastic User Interfaces", propose un *framework* conceptuel qui aide au développement des interfaces plastiques, soit en identifiant clairement la couverture fonctionnel des outils comme ARTStudio, soit en nous aidant avec la conception de nouveaux outils.

Même si ces deux modèle présentés sont déjà une grande évolution vers les conceptions des interfaces complexes exigées par les nouveaux paradigmes d'interactions, il y a encore un long chemin à parcourir et ces modèle doivent être raffinés et évoluer avec des nouvelles expériences.

4. CARTE DE CONCEPTS

L'ellipse en bleu montre les domaines couverts par les articles traités dans ce travail.



5. BIBLIOGRAFIE

Calvary, G. and Coutaz, J. *Plasticité des Interfaces: Une nécessité!*

Calvary, G., Coutaz, J., Thevenin, D. et al (2002) *Plasticity of User Interfaces: A Revised Reference Framework*.

Calvary, G., Coutaz, J. and Thevenin, D. (2001) *A Unifying Reference Framework for the Development of Plastic User Interfaces*. Proceedings of 8th IFIP International Conference on Engineering for Human Computer Interaction EHCI, May 11-13, 2001, Toronto, Canada. Lecture Notes in Computer Science, Vol. 2254, Springer-Verlag.

Calvary, G., Coutaz, J., Dâassi, O. et al. *Towards a New Generation of Widgets for supporting software plasticity: the 'Comet'*.

Calvary, G., Coutaz, J. and Thevenin, D. (2000) *Embedding Plasticity in the Development Process of Interactive Systems*. HUC 2000 Workshop on Resource Sensitive Mobile HCI.