# Architecture pour l'adaptation de Systèmes d'Information Interactifs Orientés Services

Mireille Blay-Fornarino, Vincent Hourdin, Cédric Joffroy, Stéphane Lavirotte, Sébastien Mosser, Anne-Marie Déry-Pinna, Philippe Renevier, Michel Riveill, Jean-Yves Tigli

# Architecture pour l'adaptation de Systèmes d'Information Interactifs Orientés Services

**Mireille Blay-Fornarino[*], Vincent Hourdin[*], Cédric Joffroy[*], Stéphane Lavirotte[*,**], Sébastien Mosser[*], Anne-Marie Pinna-Dery[*], Philippe Renevier[*], Michel Riveill[*], Jean-Yves Tigli[*]**

** Laboratoire I3S (Université de Nice - Sophia Antipolis - CNRS)*
*Bâtiment Polytech'Sophia – SI 930 route des Colles – B.P. 145*
*F-06903 Sophia Antipolis Cedex*
*** also IUFM Célestin Freinet - Académie de Nice*
*89, avenue George V 06046 Nice Cedex 1*

*{blay,hourdin,joffroy,lavirott,mosser,pinna,renevier,riveill,tigli}@polytech.unice.fr*

RÉSUMÉ. *Les Systèmes d'Information (SI) évoluent dans une approche orientée service qui doit être adaptable afin de prendre en compte évolutions des sources d'information, apparition/disparition de services, mais aussi les évolutions des Systèmes d'Interaction permettant le dialogue avec les SI. Nous nous intéressons ici à l'étude globale du système, en tenant compte du besoin d'adaptation dynamique simultanée, tant au niveau du Système d'Information qu'au niveau Système Intéractif. Cet article présente une architecture de contrôle permettant ces adaptations, en tenant compte des préférences des utilisateurs.*

ABSTRACT. *Information System (IS) are moving into a service oriented approach. Such approach ensure a need in application adaptibility, taking care of business evolution, services creation or disparition and interaction system allowing dialog with IS. This paper present a global study of system, focusing on dynamic evolution, at business and interaction level. We expose a control architecture allowing such adaptation, according to users preferences.*

MOTS-CLÉS : *Architecture Orientée Services, Adaptation Dynamique, Systèmes Multi-Dispositifs*

KEYWORDS: *Services Oriented Architecture, Dynamic Adaptability, Multi-Device systems*

1re soumission à *Ingénierie des Systèmes d'Information - RTSI*, le 5 décembre 2007

## 1. Introduction

De nos jours, le développement et l'utilisation des Systèmes d'Information (SI) fait face à trois défis : la multiplication des sources d'information, l'évolution technologique des dispositifs d'interaction et l'évolution des usages.

En effet, liée à la forte croissance d'Internet, force est de constater qu'il y a de plus en plus de sources d'information disponibles. Les services d'information que nous développons, répondent à nos attentes, mais s'avèrent parfois difficiles à maintenir. Le contrôle des services que nous achetons à des entreprises spécialisées nous échappe. Les services gratuits n'ont pas de pérennité et d'accessibilité garanties. Ainsi de nombreux travaux ont pour objectif de proposer des outils pour adapter les SI aux variations des sources d'information qui les composent.

Avec la multiplication des terminaux informatiques mobiles et des objets communicants dans notre vie quotidienne, nous observons l'émergence d'une informatique que Mark Weiser a nommée "Ubiquituous Computing" (Weiser, 1991) ou informatique ambiante (littéralement omniprésente) sur la base d'un constat : *" Silicon-based information technology, is far from having become part of the environment "*. De nombreuses difficultés techniques étant jour après jour surpassées, les entités physiques de notre environnement acquièrent progressivement de nouvelles capacités de communication, une existence informatique et plus concrètement, participent à de nouvelles applications logicielles (Want *et al.*, 1999). Dans le domaine des Systèmes d'Interaction par exemple, cette évolution est à l'origine de l'apparition d'interfaces tangibles (Ullmer *et al.*, 2000) et de multi-dispositifs d'entrée (Dragicevic *et al.*, 1999) pour communiquer et interagir avec l'utilisateur dans un environnement plus familier. Devant l'avènement et l'hétérogénéité des nouveaux équipements d'Interaction Homme-Machine (IHM), le Système Interactif doit permettre l'intégration de dispositifs plus insolites offrant de nouvelles techniques d'interaction (Ballagas *et al.*, 2006).

Parallèlement, devant tant de bouleversements technologiques, les usages suscitent de nombreux changements. En quelques années, nous sommes passés d'une diffusion de l'information sur support papier à sa dématérialisation, entraînant un changement de comportement des consommateurs et des producteurs d'informations mais aussi de la nécessité d'en avoir une synthèse pertinente en fonction des supports ou des tâches à exécuter. Au-delà de l'aspect technique, l'usage même est en évolution, suivant les mutations de nos modes de vie. Nous sommes de plus en plus mobiles, aussi bien dans nos loisirs que dans nos activités professionnelles. Le lieu de travail n'est plus aussi clair qu'auparavant, entre les déplacements, le télétravail, etc. En définitive, l'utilisation des SI est en mutation.

Par cette évolution des technologies, l'adaptation des systèmes doit nécessairement être abordée aussi bien au niveau du Système d'Information qu'au niveau du Système Interactif et de l'évolution des usages. La mise en correspondance de ces adaptations constitue le cœur de notre étude.

Dans cet article, nous soulignons les capacités à l'adaptation globale à partir d'une analyse de l'existant de l'adaptation à chacun de ces niveaux (section 2). A partir de

cette analyse, nous décrivons notre solution architecturale orientée services pour l'adaptation dynamique (section 3). Finalement, nous dressons un bilan de nos contributions et dessinons les contours de nos travaux futurs (section 4).

L'ensemble de nos travaux trouve une application dans SEDUITE (Blay-Fornarino *et al.*, 2007), le Système d'Information de Polytech'Nice Sophia. SEDUITE permet de diffuser les emplois du temps, les résultats aux examens, des événements organisés par le BDE, des actualités et la météo ; le tout en fonction du contexte : le portable d'un étudiant ou l'écran à côté de la machine à café ne recevront peut-être pas nécessairement la même information, ni sa diffusion sous le même format. Nos exemples s'appuient sur ce SI.

## 2. Caractéristiques des Adaptations des Systèmes d'Information Interactifs

Nous définissons un Système d'Information interactif par trois entités conceptuelles : les SI, ensemble des sources d'information disponibles, le Système Utilisateur (SU), ensemble des objectifs des utilisateurs et le Système Interactif (SINT), ensemble des ressources interactionnelles disponibles (cf. Figure 1).

Les Systèmes d'Information sont de plus en plus complexes. Aussi l'approche Web Services permet-elle d'ajouter, de supprimer ou de remplacer des services et de les orchestrer différemment selon les besoins afin de construire de nouveaux services. Ces systèmes sont auto-adaptatifs et il est important que ces adaptations puissent être notifiées (flèche Info_out de la Figure 1) et prises en compte si nécessaire au niveau des utilisateurs (flèche Use_in de la Figure 1) et/ou des dispositifs de sortie (flèche Inter_in de la Figure 1).

La multiplicité des supports physiques rend les applications de plus en plus adaptables au niveau des entrées/sorties. Aussi l'approche Web Services pour Dispositifs permet-elle la découverte de nouveaux dispositifs et les systèmes évoluent ainsi dynamiquement en termes de supports physiques. Les Systèmes Interactifs deviennent ainsi auto-adaptatifs et il est important que ces adaptations (flèche Inter_out de la Figure 1) puissent être notifiées aux utilisateurs (flèche Use_in de la Figure 1) et/ou Système d'Information (flèche Info_in de la Figure 1) si nécessaire.

La diversité des utilisateurs demande une adaptation des applications aux usages. Aussi les applications sont de plus en plus adaptables à l'aide de préférences et de paramètres soit spécifiés par le développeur, soit précisés par l'utilisateur ou soit déduits par apprentissage. Les utilisateurs introduisent alors une variabilité au niveau de l'enchaînement des tâches et de leur visualisation (flèche Use_out de la Figure 1) qu'il est peut être important de notifier au Système d'Information (flèche Info_in de la Figure 1) et/ou Système Interactif (flèche Inter_in de la Figure 1).

Cet article présente la mise en œuvre d'une architecture adaptable permettant de combiner et de faciliter la prise en compte de l'ensemble de ces adaptations sans que l'un de ces éléments de base ne soit dépendant d'un autre.

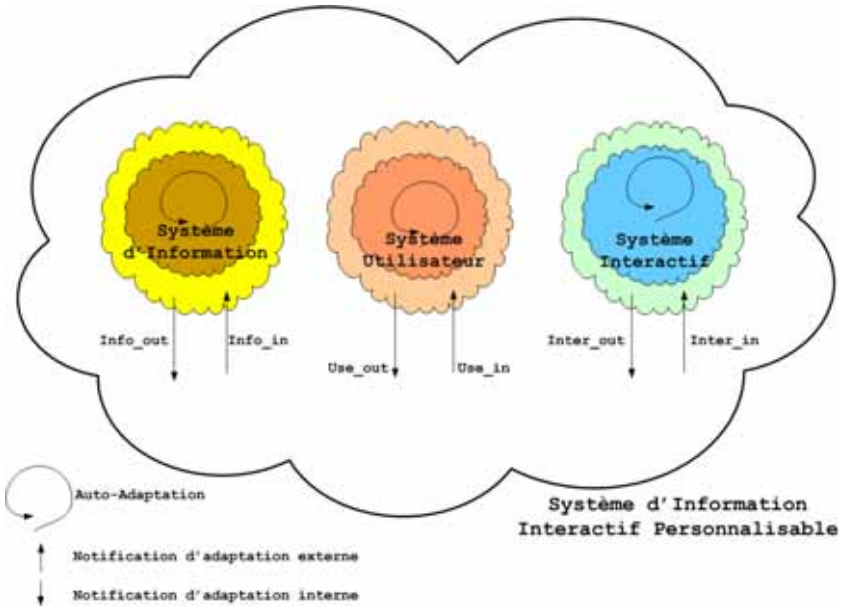


**Figure 1.** *Auto-adaptations et Notification d'adaptation au sein d'un Système d'Information Interactif Personnalisable*

Pour cela, nous présentons l'existant pour l'adaptation du SI (cf. 2.1), l'adaptation du SINT (cf. 2.2) et du SU (cf. 2.3). Une fois ces trois points exposés, nous décrivons notre solution résultante à la problématique : une architecture basée sur les Web Services que nous présentons dans la partie 3.

### 2.1. *Adaptation des systèmes d'information à base de Web Services*

Les Systèmes d'Information doivent pouvoir s'adapter rapidement en fonction des changements du marché. Les architectures orientées services (SOA pour *Service Oriented Architectures*) ont été définies pour répondre à des besoins de réutilisation et d'adaptabilité. Une architecture orientée services est un style d'architecture fondée sur la description de services et de leurs interactions (MacKenzie *et al.*, 2006). Les caractéristiques principales d'une architecture orientée services sont le couplage faible, l'indépendance par rapport aux aspects technologiques et l'extensibilité. La propriété de couplage faible implique qu'un service n'appelle pas directement un autre service ; les interactions sont gérées par une fonction d'orchestration. Il est donc plus facile de réutiliser un service puisqu'il n'est pas directement lié aux autres. L'indépendance par rapport aux aspects technologiques est obtenue grâce aux contrats d'utilisation qui sont indépendants de la plate-forme technique utilisée par le fournisseur du service. Enfin, l'extensibilité est rendue possible par le fait que de nouveaux services peuvent être dé-

couverts et invoqués à l'exécution. Plus d'informations sur ce sujet sont disponibles dans (Le Meur *et al.*, 2006). Dans cette section, nous identifions les principales adaptations que peut subir un Système d'Information en nous limitant aux architectures à base de Web Services.

Pour chaque point d'étude, nous soulignons les adaptations potentielles auxquelles les interfaces utilisateurs doivent faire face, notées $\text{SI}_i$. Ces $\text{SI}_i$ correspondent aux flèches Info_in et Info_out de la Figure 1.

#### 2.1.1. *Publication et découverte de Web Services*

UDDI (Universal Description, Discovery and Integration of Web Services) (OASIS, 2002) est le principal standard pour la publication et la découverte de services. Les Web Services exposent leur interface d'utilisation via le langage WSDL (Web Services Description Language) (Christensen *et al.*, 2001). Celui-ci assure l'indépendance vis-à-vis de la plate-forme technique utilisée par le fournisseur du service. La découverte des services sur la base de leur seule interface est difficile par programme. C'est pourquoi des travaux tels que la définition du langage OWL-S (Ontology Web Langage for Service) ou SWSO (W3C, 2005) étendent la description des services via l'utilisation d'ontologies. De même, nous trouvons des extensions de WSDL pour permettre une sélection des services en fonction de leurs propriétés de qualité de services (WSLA (Keller *et al.*, 2003)).

$\text{SI}_1$ : *Notification de création/disparition d'un service*.
Cette adaptation du SI impacte l'interface utilisateur qui évolue en fonction des services disponibles. *Par exemple, la création d'un nouveau service pour obtenir le menu du jour au restaurant universitaire peut être notifiée à l'utilisateur qui décidera ou non d'y avoir recours.*

$\text{SI}_2$ : *Obtenir un service à partir d'une description.*
Cette capacité du SI permet de construire dynamiquement des interfaces utilisateurs adaptables. *Par exemple, en cas de défaut de service, cette capacité du* SI *permet d'adapter l'interface utilisateur qui se connectera à un service équivalent.*

#### 2.1.2. *Composition des Web Services*

Pour renforcer le couplage faible entre les services, différents formalismes ont été définis pour faciliter la composition de services dont la spécification BPEL4WS qui permet de composer des services, en définissant des *orchestrations* (Khalaf *et al.*, 2003). Le résultat d'une orchestration est un nouveau service, ce qui permet une composition récursive des orchestrations. La composition de ces nouveaux services peut alors être validée par des vérifications formelles (Camara *et al.*, 2005, Martens, 2005). Grâce aux mécanismes de composition et à l'usage d'ontologies, il est possible de déterminer les services les mieux adaptés à la résolution d'une tâche (Paolucci *et al.*, 2003, Balke *et al.*, 2003). En fonction des approches, il s'agit donc de prendre en compte soit l'adaptation par l'utilisateur soit l'auto-organisation (Prehofer *et al.*, 2005).

La prise en compte de propriétés non-fonctionnelles (sécurité, traçage, cryptographie...) qui enrichissent les services est abordée via différents travaux (Charfi *et al.*, 2004, Courbis *et al.*, 2005a, Courbis *et al.*, 2005b) qui s'appuient sur le paradigme de la programmation par aspects (Kiczales *et al.*, 1997, Douence, 2004). L'utilisation de ce paradigme pour modifier les orchestrations est montrée dans les travaux d'Anis Charfi (Charfi *et al.*, 2005b, Charfi *et al.*, 2005a).

$SI_3$: *Création d'un nouveau service à partir de la définition d'une orchestration de Web Services.*

$SI_4$: *Création d'un nouveau service par fusion d'orchestrations.*
Ces capacités du SI permettent de diriger la construction de services dédiés soit en composant des services, soit en composant des compositions de services. *En fonction de l'utilisateur, l'assemblage des services de diffusion des informations est différent. Cette capacité du SI nous permet de créer des services dédiés à des groupes d'utilisateurs, ceci en fusionnant automatiquement des services eux-même composites.*

$SI_5$: *Notification d'adaptation d'un service*
L'adaptation du SI par modification de services existants peut impacter les interfaces utilisateurs. *L'introduction de sécurité sous la forme d'authentification dans le* SI SEDUITE *s'est répercuté au niveau des interfaces utilisateurs qui doivent fournir une clef avant d'émettre une requête vers le* SI.

### 2.2. *Adaptation du Système Interactif à base de Web Services pour Dispositifs*

Dans l'approche d'informatique ambiante ou "Ubiquitous Computing" décrite en introduction, nous retrouvons une multitude de dispositifs permettant d'interagir avec l'utilisateur. Il faut pouvoir s'adapter aux besoins de l'utilisateur et aux dispositifs disponibles, ceux-ci pouvant apparaître ou disparaître à tout moment.

Le Système Interactif doit donc concilier : une approche descendante (adaptation des éléments interactifs) et une approche montante (recherche, découverte, et publication de nouveaux équipements interactifs).

#### 2.2.1. *Services pour dispositifs du Système Interactif*

Nous regroupons les équipements interagissant avec l'environnement physique dans l'espace de l'utilisateur, sous le terme générique de *dispositifs d'entrée/sortie* (traduction d'"input/ouput devices") ou plus simplement de *dispositifs*. La notion de dispositif ne se limite donc pas à la seule notion de périphérique. Il peut s'agir de capteurs, d'effecteurs, d'équipements de natures diverses que l'on rencontre dans les nouveaux supports d'interactions en informatique ambiante. Les vertus précitées des approches SOA présentent alors un intérêt tout aussi conséquent pour de tels systèmes multi-dipositifs. Le projet SIRENA (Jammes *et al.*, 2005), basé sur une architecture orientée services pour des dispositifs, met en évidence, la facilité d'intégration d'un équipement, sa réutilisabilité, l'extensibilité du système à l'exécution, et l'interopérabilité entre services. Nous nous intéressons plus particulièrement aux *Web Services pour Dispositifs* (WSD), en re-

groupant sous cet intitulé l'ensemble des approches destinées à concilier les principes des Web Services et l'utilisation de dispositifs par nature contraints par des ressources physiques. Les approches plus connues en matière de WSD sont UPnP et plus récemment DPWS (Schlimmer *et al.*, 2006).

Les *spécificités* des services pour dispositifs résident alors dans les contraintes associées à de telles cibles (Hourdin *et al.*, 2006). Il s'agit souvent de *contraintes liées aux ressources* hétérogènes de chaque équipement : *déconnexions fréquentes, bandes passantes, capacités en mémoire, stockage de l'énergie, etc.* L'extension de la description de services pour dispositifs par la présence de méta-données est alors primordiale afin de renseigner les contraintes associées à l'utilisation du service. Des travaux sur les ontologies, comme (FIPA, 2002), spécifiques à la description de dispositifs et de leurs contraintes sont donc nécessaires (Bandara *et al.*, 2004) à la description complète d'un service pour dispositifs.

Les dispositifs étant le plus souvent connectés à l'environnement réel de l'application, les services associés se doivent d'offrir des caractéristiques ménageant la *réactivité* de l'application aux variations de l'environnement, ce qui n'est pas sans conséquence sur les protocoles de communication des WSD. Ces derniers définissent donc des protocoles de communication par événements (abonnement, notification) dans un contexte d'exécution asynchrone. UPnP utilise pour cela le protocole GENA (Cohen *et al.*, 1998), tandis que DPWS s'appuie sur WS-Eventing (W3C, 2006).

Enfin la notion de *localité* est omniprésente dans la disponibilité d'un service pour dispositif. Son utilisation logicielle est très souvent implicitement liée à sa proximité dans l'environnement de l'utilisateur (Pauty *et al.*, 2004). Les serveurs d'annuaire de services centralisés (UDDI) sont alors souvent difficiles à maintenir pour des dispositifs avec des déconnexions fréquentes, provoquant un surcoût de communication de maintien en cohérence des informations disponibles dans l'annuaire. De plus ils ne peuvent évaluer la proximité du dispositif avec l'utilisateur. Les solutions adoptées par les WSD reposent alors sur des mécanismes de découverte locale et pair à pair entre fournisseurs et consommateurs de services. C'est le cas pour UPnP et DPWS, avec respectivement les protocoles SSDP et WS-Discovery (Schlimmer, 2005). La notion de localité est alors pour des raisons purement techniques, liée au routage des messages et associée à l'appartenance à un même réseau local. Il apparaît donc plus naturel de conditionner la découverte d'un service au contexte. En effet l'intérêt d'un service pour dispositif dans l'environnement de l'utilisateur n'est pas seulement lié à des critères simplement géographiques. Nous pouvons par exemple conditionner la disponibilité d'un service et donc sa découverte à des plages horaires autorisées et plus généralement à des conditions contextuelles d'utilisation (Lavirotte *et al.*, 2005). Certains travaux mettent ainsi l'accent sur l'introduction du contexte dans les mécanismes de découverte de services (Kuck *et al.*, 2007).

Nous allons donc étudier les points d'adaptation du système d'information interactif aux évolutions du Système Interactif, noté $\text{SINT}_i$. Ces $\text{SINT}_i$ correspondent aux flèches Inter_in et Inter_out de la Figure 1.

SINT$_1$: *Adaptation aux évolutions du Système Interactif*
Lors de l'apparition ou la création d'un nouveau service pour dispositif, l'utilisateur doit pouvoir indiquer s'il compte le rajouter à l'ensemble des services pour dispositifs qu'il utilise. De même, il doit pouvoir supprimer à sa guise un service pour dispositif de la liste de ceux qu'il utilise.

SINT$_2$: *Adaptation aux conditions d'utilisation du Système Interactif*
L'adaptation du système d'information interactif doit prendre en compte des prérequis fournis par l'utilisateur pour l'utilisation d'un service pour dispositif ou son adaptation. *Par exemple la non-utilisation d'un afficheur public si l'utilisateur n'est pas seul ou encore la suppression du mode sonnerie du téléphone mobile dans un environnement bruyant.*

### 2.2.2. *Adaptation du Système Interactif*

Dans le domaine de l'IHM cette problématique est étudiée pour conférer aux IHMs la propriété de plasticité (Calvary *et al.,* 2004). L'adaptation au contexte d'usage (cf. 2.3) au niveau (2),(3) et (4) a un impact indirect sur le Système Interactif, comme nous le verrons dans le paragraphe 2.3.1 et dans le cadre de la référence CAMELEON. Il s'agit d'une adaptation induite par les niveaux supérieurs qui se traduit alors par la création de nouveaux services d'interaction par composition et/ou adaptation de services d'interaction existants grâce à la programmation par aspects (voir SINT$_4$ et SINT$_5$).

Le premier enjeu pour l'adaptation repose donc sur la création à l'exécution de nouveaux services pour dispositifs à partir de services pour dispositifs recherchés et découverts dynamiquement dans l'environnement de l'utilisateur et de services pour dispositifs composés dynamiquement pour créer des services d'interaction de haut niveau.

SINT$_3$: *Création d'un nouveau service pour dispositif par composition*
L'adaptation d'un système multi-dispositifs implique sa capacité à rechercher, découvrir et composer dynamiquement des services pour dispositifs pour fournir des services de plus haut niveau.

Le second enjeu réside dans la capacité d'adaptation des services pour dispositifs existants du Système Interactif.

SINT$_4$: *Adaptation d'un service d'interaction par application d'aspects*
L'adaptation d'un système multi-dispositifs implique sa capacité à modifier dynamiquement des services pour dispositifs selon différentes préoccupations.

Le troisième enjeu concerne l'auto-adaptation du Système Interactif, apparu plus récemment comme une conséquence directe de la multiplicité et la variabilité des dispositifs d'interaction généralement présents dans le contexte d'un utilisateur. Il est apparenté au niveau (1), celui de l'interface finale, puisqu'interne au SINT. L'auto-adaptation prise en charge par le Système Interactif n'adresse donc qu'en partie le problème de plasticité. Il s'agit d'une adaptation au contexte que nous appelons *contexte d'exécution* (Tigli *et al.,* 2006) directement liée à l'évolution de l'infrastructure du système et par nécessité, plus proche de la problématique des applications sensibles aux ressources .

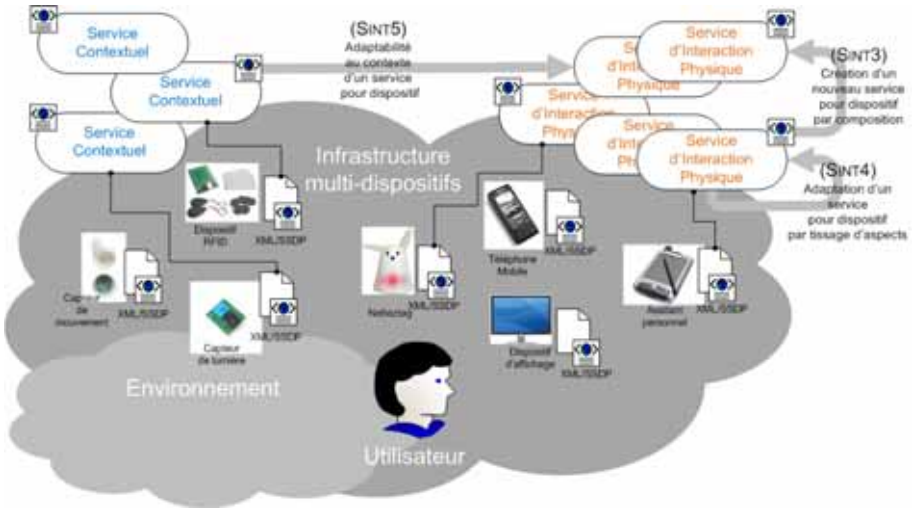


**Figure 2.** *Principes d'adaptation du Système Interactif*

SINT$_5$: *L'adaptativité d'un service pour dispositif*
L'adaptation du Système Interactif implique sa capacité à percevoir et réagir aux variations de son contexte. Pour cela, le Système Interactif doit recueillir des informations contextuelles et s'adapter. Nous détaillerons cette partie dans la section 3.4.

L'ensemble des principes pour l'adaptation à l'évolution du Système Interactif sont regroupés dans la Figure 2.

### 2.3. *Adaptation du Système Utilisateur*

La bivalence entre la partie fonctionnelle et la partie interactive est un des principes fondamentaux de la recherche en IHM. Cette décomposition est illustrée par le modèle architectural Arch (Bass *et al.*, 1992) qui offre une décomposition fonctionnelle d'une application interactive en séparant l'interface utilisateur des fonctionnalités métiers (Noyau Fonctionnel – NF). Ainsi, le Contrôleur de Dialogue (CD) est le niveau conceptuel introduit pour gérer le dialogue :

– Du point de vue de l'utilisateur, il s'agit d'articuler, d'organiser et de réorganiser le déroulement de l'interaction. En d'autre termes, il s'agit de prendre en compte les préférences exprimées par l'utilisateur dans le SU. Aussi dans cet article, nous concentrons nos travaux sur cette prise en compte au niveau architectural. De ce fait, nous n'avons pas exploité les résultats de la recherche autour des SU (ontologie, profil, apprentissage, etc.).

– Du point de vue informatique, il s'agit de relier les fonctions aux interactions. C'est pourquoi nous voyons le CD, en l'élargissant à ses adaptateurs, comme un interlocuteur anonyme du NF et de l'interaction (ou Présentation Physique – PP).

Dans le domaine des IHMs, dont est issu le CD, l'aspect fonctionnel est cloisonné de sorte que les travaux sont orientés côté interaction. Dans la littérature IHM, l'adaptation du CD est étudiée en relation avec l'adaptation de la Présentation Logique (PL), dans le cadre de la plasticité (Thévenin, 2001). Par définition, la plasticité des IHMs dénote la capacité d'une IHM à s'adapter à son contexte d'usage dans le respect de son utilisabilité. Le contexte d'usage est alors défini par le triplet <utilisateur, environnement, plate-forme>.

#### 2.3.1. *Adaptation du Contrôleur de Dialogue : existant*

La prise en compte de l'utilisateur dans l'adaptation est décrite en 4 niveaux dans le cadre de référence CAMELEON[1]. Pour un contexte d'usage donné, ces niveaux sont par ordre de niveaux décroissants d'abstraction : (4) les tâches (buts et sous buts de l'utilisateur) et le domaine (objets manipulés) ; (3) l'interface abstraite (expression de l'interaction indépendamment des modalités d'interaction) ; (2) l'interface concrète (expression de l'interaction indépendamment de la plate-forme d'exécution) ; (1) l'interface finale (l'interface opérationnelle). Les possibilités d'adaptation se font par changement d'un contexte d'usage à un autre, par combinaison de trois relations : la translation (changement de contexte à un même niveau), l'abstraction (passage vers un niveau d'abstraction supérieur) et la réification (passage vers un niveau d'abstraction inférieur). Nous présentons dans ce cadre de référence les travaux menés en IHMs sur l'adaptation du CD et de la PL, selon trois axes d'études : les architecture à agents, les Langages de Description d'Interface Utilisateur (LDIU) et l'Ingénierie Dirigée par les Modèles (IDM).

Les architectures à agents des IHMs correspondent à une approche logicielle modulaire. Historiquement, nous pouvons citer les approches MVC (Reenskaug, 1979) ou PAC (Coutaz, 1987). Aujourd'hui, il existe des approches conçues pour la plasticité et donc l'adaptation : AMF (Tarpin-Bernard *et al.*, 1999) et Comet (Demeure *et al.*, 2006). Chacune de ces approches adopte plusieurs facettes, et l'adaptation consiste alors à changer une facette (orientée présentation) pour une autre. L'adaptation porte aussi sur une reconfiguration de la structure des agents ce qui correspond à une modification de l'enchaînement des tâches utilisateurs.

Les Langages de Description d'Interface Utilisateur (souvent basés sur XML) sont une autre approche pour adapter les IHMs. Ces langages comportent différents niveaux d'abstractions correspondant aux niveaux du cadre CAMELEON. Ils incluent les mécanismes d'abstraction et de réification. Nous pouvons citer les travaux du W3C sur le *device independance* (W3C, 2001) et des langages comme UsiXML[2]. Ce dernier, créé avec le cadre de référence CAMELEON, illustre l'ambition des LDIU : (i) association tâche – élément d'interaction (par catégorie : navigation, choix, etc.), (ii) description abstraite de l'interface, (iii) description concrète de l'interface et (iv) moteur de rendu.

1. Projet européen CAMELEON, http ://giove.cnuce.cnr.it/cameleon.html
2. http ://usixml.org/

La dernière approche pour adapter une application interactive consiste à intégrer les modèles conceptuels aux applications afin d'en regénérer des parties en cas de changement de contexte. Deux sortes d'application de l'IDM à la plasticité des IHMs sont identifiables. (i) Dans (Sottet *et al.*, 2006), un modèle est associé à chaque niveau du cadre CAMELEON pour chaque contexte d'usage et donc l'adaptation se fait par transformations successives de modèles. (ii) Dans (Hariri *et al.*, 2006), elle est réalisée par l'association de patrons de conception et l'utilisation de bases de connaissances afin d'associer contexte capturé, modèles et patrons .

Ces trois approches pour l'adaptation du CD permettent de couvrir les niveaux (4), (3) et (2) du cadre CAMELEON. Notons également que ces approches ne sont pas incompatibles, par exemple dans (Martinez-Ruiz *et al.*, 2006) les auteurs combinent les langages de description et l'IDM.

### 2.3.2. *Adaptation du Contrôleur de Dialogue : constats*

Les approches de l'adaptation en IHM décrites précédemment sont centrées utilisateur et orientées conception. Au delà des qualités de ces démarches « descendantes » (*top-down*), nous relevons les manques suivants :

**Manque de couverture de l'adaptation du NF et du SINT.** En effet, nous considérons que les pieds de l'arche, le NF, SI dans notre cas, et la PP, SINT dans notre cas, ont une existante intrinsèque, indépendamment de l'utilisation qui en est faite. En conséquence, l'adaptation prenant en compte l'utilisateur et ses objectifs, à l'exception de quelques préférences liées aux dispositifs ou aux fonctionnalités, ne peut avoir lieu que dans le CD et la PL. Les travaux d'adaptation se concentrent donc sur les niveaux (4), (3) et (2) du cadre CAMELEON. Cependant, nous souhaitons dépasser le clivage IHM – NF. Bien sûr la modularité apporte la flexibilité et la réutilisation, cependant un clivage trop fort coupe la partie interactive d'informations pertinentes pour l'utilisateur et réciproquement. Ce qui nous conduit au point suivant.

**Manque de souplesse dans l'adaptation.** Même si les adaptations étudiées sont déclenchées par une analyse du contexte, elles sont traitées à partir du CD pour se projeter jusqu'à l'interface finale. Or, nous constatons que les adaptations peuvent se produire indépendamment dans le SI ou le SINT. Il est alors nécessaire de les propager via le CD.

**Manque de réutilisation de l'existant.** L'approche descendante suppose un contrôle sur les différentes entités manipulées. Le résultat se trouve cloisonné dans le cadre de travail fixé, ne permettant pas la prise en compte d'éléments existants non conformes au modèle attendu.

Par ailleurs, les Systèmes d'Information ont la particularité que les informations sont partagées et les clients répartis. Les approches langages de description ou IDM s'avèrent alors trop lourdes pour les faire fonctionner à l'exécution sur des dispositifs légers (téléphone, PDA, réseau sans fil "vite" saturé), tandis que l'approche à agents nécessite un développement spécifique pour le traitement des données et le protocole de communication rendant difficile une adaptation non anticipée. Il faut donc trouver

un compromis entre la modularité, la distribution et la souplesse de déploiement pour l'adaptation dynamique.

Résumé en une phrase, la problématique que nous étudions est l'exploration « d'une architecture logicielle permettant la réutilisation de l'existant et autorisant l'adaptation et sa propagation de chacun des trois Systèmes d'Information, Utilisateur et Interactif indépendamment des deux autres. ». Aussi défendons nous l'idée que l'approche orientée services est la base d'une solution à cette problématique.

## 3. Architecture pour un Système d'Information Interactif adaptable et mise en œuvre

Dans notre étude de l'existant, nous avons souligné les points de jonctions ($SI_i$ et $SINT_i$) correspondant aux notifications d'adaptation de la Figure 1. Dans cette section, nous exploitons ces résultats pour présenter un modèle d'architecture logicielle inspiré du modèle Arch en l'étendant en une architecture tripartite non-orientée (SI, SU et SINT) coordonnée par un contrôleur de dialogue adaptable (CDA).

Au cœur du CDA nous plaçons une représentation des conséquences des adaptations sur les besoins utilisateurs sous forme de modèle de tâches. A partir d'un modèle de tâches initial, issu du SU et qui est l'expression des différents objectifs (tâches) et objectifs intermédiaires (sous-tâches) que devront atteindre un utilisateur pour réaliser un but final, nous faisons évoluer le modèle de tâches selon les adaptations notifiées par les trois systèmes (SI, SU et SINT). Le modèle de tâches est explicité et décoré afin qu'il puisse être évalué en vue d'une exécution par le système. Nous avons choisi d'exprimer momentanément l'ordonnancement des tâches avec un formalisme simple : HTA (Annett *et al.*, 1967). Ce dernier décrit les objectifs utilisateurs sous la forme d'un arbre de tâches décrivant séquentialité, alternative et rebouclage. La racine de l'arbre est la tâche globale, celle qui motive l'utilisateur. Les feuilles d'un arbre sont les tâches élémentaires qui se traduisent par des actions physiques de l'utilisateur. Ce formalisme de description de modèle de tâche n'est certes pas aussi riche que d'autres incluant des relations temporelles, mais correspond à nos premières ambitions implémentatoires.

La Figure 3 représente l'arbre de tâches HTA correspondant à une utilisation de l'emploi du temps d'une personne afin de la localiser à un moment donné (passé, présent ou futur).

A partir de cet arbre de tâches, nous illustrons son évolution consécutivement à des notifications d'adaptation provenant d'un des systèmes présentés à la Figure 1 :

**Adaptation propagée par le SI (Info_out sur la Figure 1).** *La tâche "choix de la personne" dépend du service fonctionnel disponible. Lorsqu'il se limite à une simple requête à partir du nom de la personne dans une base de données, l'interface associée ne doit permettre que de saisir le nom recherché. A l'inverse, si le service de recherche permet de faire des requêtes avancées, permet de restreindre l'ensemble des éléments à explorer, etc., alors l'interface associée devra*

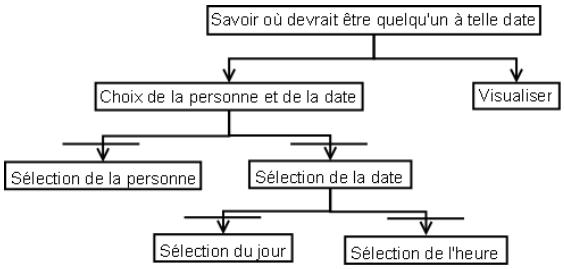


**Figure 3.** *Exemple d'arbre de tâches*

*être plus riche.* La découverte ou disparition de services d'information et leur interface impacte donc sur l'arbre de tâches en explicitant une des branches.

**Adaptation propagée par le SINT (Int_out sur la Figure 1).** *Outre les choix de conception, la décomposition plus ou moins précise de la sous-tâche "sélection du jour" (jusqu'à la sélection du jour, du mois et de l'année) est amenée à évoluer en fonction des événements propagés par le Système Interactif. Par exemple, si pour choisir la date un service interactif basé sur des champs textuels est disponible, la sous-tâche "sélection du jour" est alors une tâche élémentaire de saisie d'informations. Si un service interactif basé sur un calendrier mensuel graphique avec des éléments de navigation (jour / mois / année) est proposé, alors la sous-tâche "sélection du jour" se complexifie car elle intègre une tâche élémentaire de navigation.* La découverte ou disparition d'éléments interactifs impacte donc le sous arbre correspondant de l'arbre de tâches.

**Adaptation choisie par le SU (Use_out sur la Figure 1).** *La visualisation du résultat de la recherche (lieu où est la personne) peut être de nature très différente : un simple texte(adresse postale ou coordonnées), une carte, etc. Chaque visualisation correspond à une idée de représentation (interface abstraite). Si l'utilisateur peut choisir cette représentation, le* CDA *doit alors gérer les données en vue de leur représentation.* Il n'y a pas forcément d'impact direct sur l'arbre de tâches, mais sur le traitement associé aux sous tâches. L'arbre de tâches est décoré en conséquence afin de faciliter son évaluation.

En résumé, le rôle du CDA que nous proposons est de faire le lien entre une tâche (objectif de l'utilisateur), les fonctionnalités, le rendu et les entrées utilisateurs. Les éléments à adapter sont : le modèle de tâches initial ou l'association tâche-fonctionnalité ou l'association tâche-rendu ou l'association tâche-entrées utilisateurs (par ajout de méta données sur l'arbre). Nous proposons dans la sous-section (3.1) : un modèle d'architecture permettant d'agir selon ces quatre axes (3.1.1) et son implémentation dans SEDUITE (3.1.2).

### 3.1. *Arche tripartite non-orientée adaptable*

#### 3.1.1. *Modèle d'Architecture*

La Figure 4 décrit une extension du modèle Arch sur lequel plusieurs éléments d'adaptation sont associés. Cette Arche est composée d'un Contrôleur de Dialogue Adaptable (CDA) qui coordonne les adaptations notifiées par trois systèmes : SI (correspondant au Noyau Fonctionnel du modèle Arch), SU (non explicitement exprimé dans Arch) et SINT (correspondant à la Présentation Physique du modèle Arch). Les trois pieds sont des entités indépendantes sur lesquelles se placent les éléments de l'Arche adaptable pour faciliter la mise en œuvre d'un système d'information interactif adaptable. Dans notre modèle chaque pied est multiple. En effet, l'interaction s'exprime par de multiples couples - service pour dispositif / Adaptateur correspondant au classique couple Présentation Logique / Présentation Physique. Cette nouvelle décomposition provient du fait que le Système Interactif est composé d'un ensemble non figé de services pour dispositifs différents auxquels sont associés des adaptateurs facilitant la transmission des données entre le CDA et les services fournis.

Le couplage Adaptateur du Noyau Fonctionnel / Noyau Fonctionnel correspond quant à lui à plusieurs couples service d'information / Adaptateur. En effet le Noyau Fonctionnel se définit par un ensemble évolutif de Web Services dédiés. A chaque Web Service d'information correspond un adaptateur approprié facilitant la transmission des données entre le CDA et les services fournis.

Nous avons introduit la capacité de transmettre les informations entre le SU et le CDA en suivant la même décomposition : service utilisateur / Adaptateur. Ces informations sont illustrées dans la Figure 4 sous forme d'éléments de l'adaptation appelés dans la suite *préférences*. Ces préférences sont conceptuellement associées à tous les niveaux de l'Arche car l'adaptation au niveau du SU permet de tout personnaliser : SI, CDA et SINT.

Dans notre modèle, les adaptateurs font partie de l'Arche adaptable. Ils ont en effet pour rôle de faciliter le flot de données entre les pieds et le Contrôleur de Dialogue Adaptable.

A partir des préférences utilisateurs arrivant du SU [3], le CDA prend les décisions de mise en forme et de répartition des informations. Devant composer avec plusieurs sources d'information (les différents pieds de l'Arche, côté fonctionnel) et devant également composer avec plusieurs dispositifs d'interaction (les différents pieds de l'Arche, côté interaction), le CDA a explicitement un double rôle de fusion/fission de données : pour les informations et pour les interactions. Pour les données fonctionnelles, le CDA réorganise les données de plusieurs sources d'information (fusion) et les restructure en fonction de leurs usages (formattage). Ensuite, le CDA doit redistribuer les informations préparées aux différents dispositifs de sortie concernés. Pour les données interactionnelles, le CDA recompose le fil de l'interaction (fusion de haut niveau) pour rediriger les

3. L'utilisateur en interagissant avec une méta-interface peut formuler des préférences quant aux informations qu'il désire recevoir et leur rendu

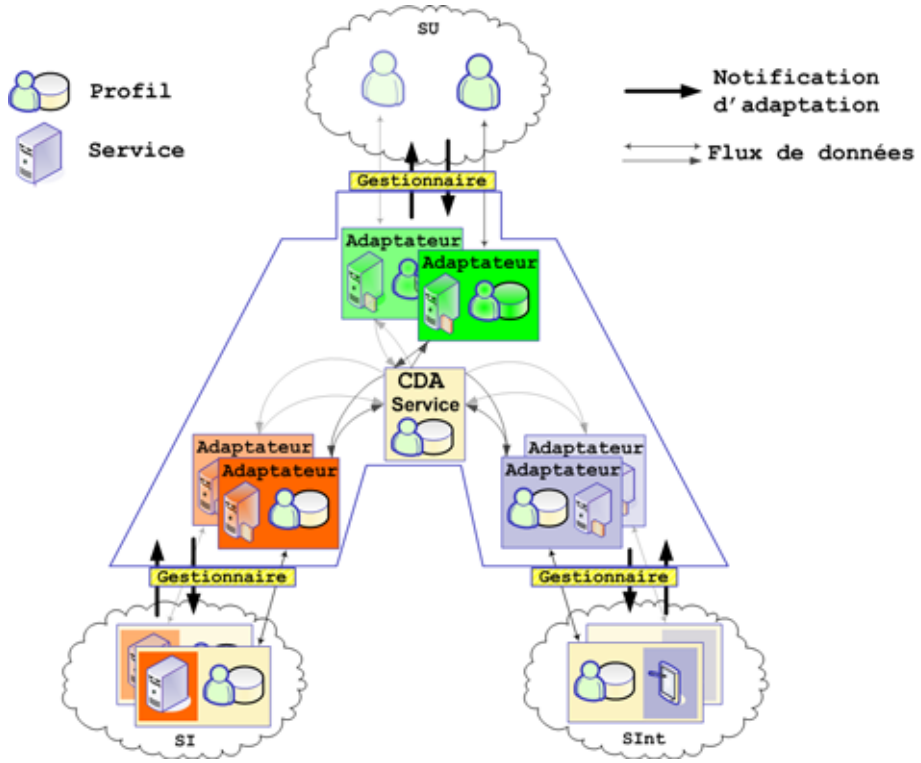


**Figure 4.** *Modèle d'une Arche adaptable*

informations perçues vers les éléments du Noyau Fonctionnel concernés. Le CDA a également pour fonction d'informer le SU des différentes adaptations qui se sont produites au niveau du SI et du SINT en modifiant l'ensemble des choix de services d'information et de services pour dispositifs à disposition.

Les adaptations initiées par une entité externe au CDA sont reçues par les GESTIONNAIRES la Figure 4. Ce sont les éléments de connexion entre l'Arche adaptable et les trois causes d'adaptation qui sont indépendantes : SI, SU et SINT.

En 3.1.2, nous décrivons les bases de l'implémentation de notre modèle. Puis les gestionnaires sont présentés dans les sous-sections 3.2, 3.3 et 3.4.

### 3.1.2. *Implémentation du Contrôleur de Dialogue Adaptable*

Une implémentation du Contrôleur de Dialogue Adaptable décrite dans la Figure 5 repose sur une décomposition en services des différentes fonctionnalités attendues du CDA. L'implémentation en services facilite l'évolution de l'architecture et l'adaptation du CDA en orchestrant différemment les services selon le type d'adaptation visé. L'im-

plémentation a été réalisée à base de Web Services en .Net C#. L' assemblage actuel est décrit par la Figure 5. Il traite d'un sous ensemble des adaptations. Il gère essentiellement la prise en compte des adaptations issues du SU et de leur répercussion au niveau SI et SINT. Actuellement le SINT n'est exploité qu'en sortie. Les services actuellement implémentés sont les suivants :

– un service de gestion de préférences. Ce service permet d'ajouter, de modifier et d'exploiter l'ensemble des préférences (services et paramètres, représentation, dispositifs, police d'écriture, ...).

– un service de tâches qui permet de modifier, de décorer et d'exploiter le modèle de tâches de base. Actuellement il s'occupe de déterminer pour un usage, grâce aux préférences, le modèle de tâches correspondant.

– un service d'évaluation qui récupère l'ensemble des informations nécessaires à partir d'un modèle de tâches. Concrètement il appelle les méthodes sur les services et renvoie la concaténation de tous les résultats.

– un service de transformation qui prépare le rendu vers le Système Interactif. Il produit une IHM abstraite à partir d'informations et de préférences (d'affichage et du choix de la présentation). Les présentations sont construites à partir des données XML que le CDA reçoit des Web Services présents dans le Noyau Fonctionnel. Ces données sont enrichies afin de prendre en compte les préférences utilisateurs. Selon les préférences de l'utilisateur (base de données "feuilles de style XSL" sur la Figure 5), le système applique la feuille de style adéquate pour effectuer la transformation en XAML (eXtensible Application Markup Language).

– un service de composition qui assemble un ensemble de rendus (interfaces abstraites).

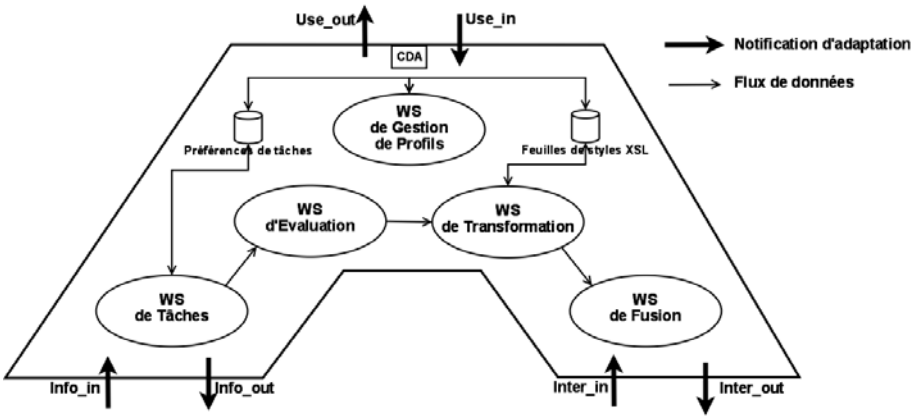


**Figure 5.** *Assemblage des Web Services implémentant le* CDA

Nous avons choisi XAML comme premier langage d'expérimentation d'interface abstraite. Le langage XAML a été développé par *Microsoft* et est une composante du

*Framework .Net 3.0.* Par la suite, pour changer de langage d'interface abstraite, il suffit de rajouter les feuilles de style prenant en compte les nouveaux formats (tel que UsiXML, Flex[4] ou OpenJFX[5]) afin de cibler les dispositifs mobiles.

### 3.2. *Gestionnaire de transmission d'adaptation* SU-CDA

Notre modèle d'architecture permet à l'utilisateur de personnaliser aussi bien les dispositifs, les informations que le CDA. L'ensemble des adaptations utiles étant remontées au SU, l'utilisateur peut changer ses préférences en fonction de l'évolution du SI et du SINT. Il peut interagir sur le système complet et choisir les informations qui l'intéressent, la représentation que vont avoir les informations récupérées, etc. Ainsi l'adaptation en provenance du SU peut-elle être répercutée à tous les niveaux. De cette façon, l'utilisateur personnalise le système d'information interactif. Ces personnalisations se déclinent sous forme de plusieurs types de préférences qui correspondent à la flèche Use_in de la Figure 1 :

1) celles concernant le paramétrage des services d'information et leur possible orchestration (niveau SI). *L'utilisateur peut préciser vouloir la météo de Nice, indiquer qu'il est un étudiant en dernière année et vouloir en conséquence l'emploi du temps de sa promotion, etc.*

2) celles concernant le filtrage des informations (niveau Adaptateur du SI). *L'utilisateur peut ne vouloir que la météo du matin et celle du soir et ignorer ainsi les informations complémentaires fournies par le service de météo.*

3) celles concernant l'enchaînement des tâches (niveau CDA). *Si l'utilisateur doit prendre connaissance de plusieurs informations, il peut spécifier l'ordre de consultation de celles-ci.*

4) celles permettant de choisir une mise en page parmi plusieurs (niveau CDA). *L'utilisateur peut préciser qu'il souhaite un rendu de l'emploi du temps style « aéroport » (horaire d'embarquement des avions) ou style « planning ».*

5) celles permettant de désigner les dispositifs de sortie préférés (niveau Adaptateur du SINT). *L'utilisateur peut préciser qu'il souhaite recevoir ses notes sur son téléphone portable et voir l'emploi du temps sur l'écran plasma de l'entrée de son école.* NB *: nous ne traitons pas explicitement des conflits d'usage sur les dispositifs publiques outre les contraintes d'accès aux dispositifs eux-mêmes.*

6) celles concernant la forme de l'affichage (*fonte, couleur, etc.*) (niveau SINT).

Les préférences sont enregistrées dans des bases de données. Leur répartition conceptuelle est illustrée sur la Figure 4. Pour des facilités d'implémentation ces bases sont regroupées sur un même serveur, accédé par le SU, le CDA et les adaptateurs. Le SU se limite à l'heure actuelle à une simple saisie des préférences via un client léger (page ASP).

4. http ://www.adobe.com/fr/products/flex/
5. https ://openjfx.dev.java.net/

### 3.3. *Gestionnaire de transmission d'adaptation* SI-CDA

#### 3.3.1. *Modèle du gestionnaire*

Ce gestionnaire peut être vu comme une interface logicielle permettant au CDA d'être observateur et pilote du SI. Ce gestionnaire correspond à la prise en compte des $SI_i$ dégagés au paragraphe 2.1. Selon les capacités du SI observé, le gestionnaire est plus ou moins complet et dirige et réagit aux adaptations autorisées par le SI. Nous listons dans ce paragraphe les adaptations que le CDA peut gérer en fonction des $SI_i$. Ces adaptations sont guidées par les adaptations qui proviennent (i) soit du SI, (ii) soit des usages. Nous soulignerons celles qui sont prises en compte par les services du CDA implémentés dans la version actuelle de SEDUITE.

**Adaptations en provenance des SI (Info_out sur la Figure 1)**

– *Adaptation par création d'un nouveau service au niveau du* SI ($SI_1$). Ces adaptations sont provoquées soit par la création d'un nouveau Web Service (cf. 2.1.1) soit par la définition d'une orchestration de Web Services. Cette adaptation est prise en charge par l'implémentation actuelle mais n'est pas illustrée dans l'orchestration des services du CDA afin d'alléger la Figure 5. Ce nouveau service doit être ajouté à la liste des services proposés dans le SU. Dans le cas d'une orchestration ou d'une fusion d'orchestrations, il est important de préciser les services élémentaires qui composent le service résultant, afin que le CDA puisse construire l'IHM abstraite à partir des IHMs abstraites de base.

– *Adaptation par enrichissement d'un service* ($SI_5$). Lors de l'enrichissement d'un Web Service, il est important au niveau du CDA de répercuter au niveau du SU la partie usage à partir des enrichissements. De plus si cela implique une adaptation de l'interface d'utilisation d'un service en fonction des enrichissements de services, le CDA doit à la fois accéder à l'information et la restituer.

**Adaptations en provenance des usages (Info_in sur la Figure 1)**

– *Adaptation par équivalence de Web Services* ($SI_2$). Cette adaptation peut être envisagée si le SI a la capacité de fournir un service à partir d'une description sémantique. En cas de défaut de service, cela permet d'offrir un service équivalent. Cette fonctionnalité est prise en charge dans l'implémentation du CDA.

– *Adaptation par demande d'orchestration*($SI_3$, $SI_4$). Dans la lignée de l'IDM appliquée à l'IHM cf. section 2.3, le CDA peut parvenir à détecter un patron récurrent dans l'enchaînement des tâches. Dans ce cas, il demande au SI de construire un Web Service adapté, résultant d'une nouvelle orchestration.

#### 3.3.2. *Plate-forme Adore et liaison entre le gestionnaire de transmission et le* SI

La plate-forme Adore supporte l'adaptation dynamique des applications à base de Web Services en s'appuyant sur le formalisme des orchestrations (plus d'informations dans (Joffroy *et al.*, 2007)). Un Web Service dit "serveur d'orchestrations" sert de relais avec le gestionnaire de dialogue du côté SI. La Figure 6 présente son interface.

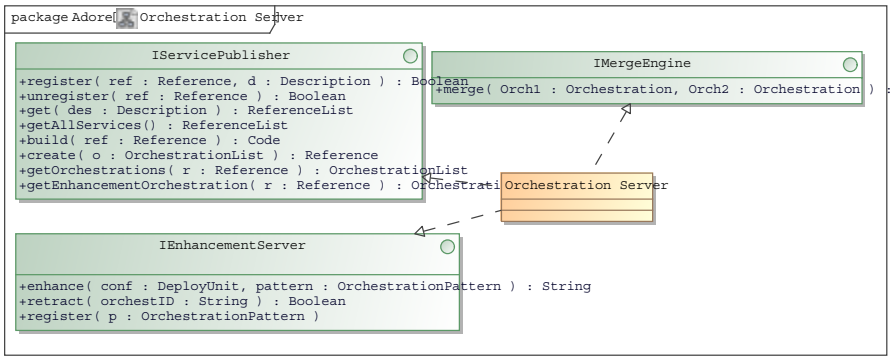


**Figure 6.** *Serveur d'Orchestrations*

L'interface *IMergeEngine* encapsule la capacité de la plate-forme à construire une nouvelle orchestration à partir de deux orchestrations (Nemo *et al.*, 2007). L'interface *IEnhancementServer* définit les opérations de support à l'enrichissement des Web Services. L'interface *IServicePublisher* présente l'ensemble des opérations permettant de définir et obtenir des Web Services contrôlables. A ce jour, la description est limitée au WSDL du service et la sélection d'un service est uniquement basée sur le nom du service.

La notification de création ou d'enregistrement d'un nouveau service ($SI_1$) est prise en charge par une orchestration d'enrichissement entre le serveur d'orchestrations et le CDA. La création d'un nouveau service à partir de la définition d'une orchestration de Web Services ($SI_3$) ou par fusion d'orchestrations ($SI_4$) est supportée par le serveur d'orchestrations. Il autorise également l'enrichissement d'un service par rajout d'activités ($SI_5$). Les informations relatives aux compositions de Web Services et à leur enrichissement sont accessibles dans le formalisme des orchestrations.

### 3.4. *Gestionnaire de transmission d'adaptation* SINT-CDA

#### 3.4.1. *Modèle du gestionnaire*

Ce gestionnaire peut être vu comme une interface logicielle permettant au CDA d'être observateur et pilote du SINT . Ce gestionnaire correspond aux $SINT_i$ dégagés au paragraphe 2.2. Selon la nature du Système Interactif observé, le gestionnaire est plus ou moins complet et permet de suivre les adaptations autorisées. Nous listons dans ce paragraphe les adaptations que le CDA peut gérer en fonction des $SINT_i$. Ces adaptations proviennent soit du SINT, soit des usages.

**Adaptations en provenance du Système d'Interaction (Int_out de la Figure 1)**

– *Adaptation à la découverte de nouveaux* WSD ($\text{SINT}_1$, $\text{SINT}_3$) : il peut s'agir d'un dispositif élémentaire ou d'un nouveau service pour dispositifs de plus haut niveau construit par composition. Dans les deux cas, le CDA doit propager l'apparition de ces services pour dispositifs au SU afin que l'utilisateur puisse les utiliser selon ses préférences en fonction de leur description. Cette fonctionnalité est actuellement prise en charge par le CDA même si l'orchestration des services correspondante n'est pas visualisée dans la Figure 5.

– *Adaptation d'un service d'interaction* ($\text{SINT}_4$, $\text{SINT}_5$) : si un service d'interaction s'est enrichi par application d'Aspects d'Assemblage (AA) ou par auto-adaptation, il est important de notifier les nouvelles fonctionnalités afin d'en référer le SU. La programmation par AA définie dans (Tigli *et al.*, 2006). reprend les grands principes de la programmation par aspects (*cf.* 2.1.2), en les adaptant aux modifications des assemblages de composants. Elles sont utilisées pour modifier le comportement du service résultant d'une composition de services par assemblage de composants.

**Adaptations en provenance des usages (Int_in de la Figure 1)**

– *Répercussion de contraintes utilisateurs sur les dispositifs* ($\text{SINT}_2$) : l'utilisateur peut par exemple demander le changement du mode d'un dispositif. Cette fonctionnalité est actuellement prise en charge par le CDA, elle fait partie des préférences que peut exprimer un utilisateur via le SU.

– *Demande d'enrichissement des dispositifs* ($\text{SINT}_4$) : lorsqu'un utilisateur fait toujours des choix de combiner des dispositifs afin d'en générer un de plus haut niveau, le CDA peut demander de le proposer comme un dispositif à part entière sous forme d'une orchestration de WSD (*i.e.* un AA).

### 3.4.2. *Plate-forme WComp et liaison entre le gestionnaire de transmission et le* SINT

WComp est une plate-forme à composants pour le développement rapide de prototypes d'applications multi-dispositifs qui doivent évoluer dynamiquement avec leur environnement d'exécution (Cheung-Foo-Wo *et al.*, 2007) et s'adapter à leur contexte (Tigli *et al.*, 2006). Elle permet de créer de nouveaux services pour dispositifs par composition dynamique de Web Services pour Dispositifs et classiques existants.

L'architecture de WComp est constituée de deux entités majeures : les *containers* et les *designers*. Un *container* gère la partie opérationnelle de la plate-forme par assemblage dynamique de composants logiciels communicants par événements. Les composants gérés par le *container* sont de différentes catégories : les composants logiciels, les sondes et les composants "mixtes". Une sonde permet soit d'exporter les événements d'adaptation vers le CDA soit d'en importer les requêtes. Un composant "mixte" est le proxy d'un Web Service pour Dispositif de l'infrastructure, donc un singleton d'accès à ce dispositif. Son apparition et disparition se fait automatiquement en fonction de celle du Web Service pour Dispositif (remontées d'informations pour $\text{SINT}_1$ et $\text{SINT}_2$).

La plate-forme WComp permet la création de WSD composites, pouvant être réutilisés comme composants mixtes dans un nouvel assemblage de composants. Notre plate-forme permet donc la mise en place d'une hiérarchie de services composites ($\text{SINT}_3$).

*Un dispositif de pointage peut par exemple être créé à partir d'un joystick sans fil et d'une liste d'affichage de choix exportée avec un* WSD.

Le *designer* de composants "mixtes" et le *designer* par AA permettent une manipulation de l'assemblage plus élaborée. Le *designer* par AA permet après sélection des aspects leur application de modifier en conséquence l'assemblage de composants du service concerné. L'adaptation d'un service pour dispositif composite peut être opérée par sélection et application d'un certain nombre d'aspects d'assemblage (SINT$_4$). *On peut vouloir modifier un dispositif composite pour changer sa modalité d'interaction, par exemple passer d'une sortie affichée à une sortie parlée ou imprimée.*

La plate-forme WComp est donc particulièrement adaptée pour gérer ici l'adaptativité du Système Interactif. Dans le cas où le déclenchement de l'adaptation est provoquée par des conditions contextuelles gérées par des services construits dans ce but, nous pourrons parler d'adaptativité ou d'auto-adaptation du service concerné du Système Interactif, au niveau (1) dans la référence CAMELEON (SINT$_5$).

## 4. Bilan et perspectives

Nous avons présenté une architecture orientée services pour l'adaptation des systèmes d'information interactifs. Prenant en compte cette adaptation à chaque niveau de notre Arche tripartite, notre contribution est leur mise en correspondance et leur propagation pendant la phase d'exécution, et non pas lors de la conception comme dans les solutions globales de conception semblables à celles décrites dans (Champalle *et al.,* 2006). Notre modèle a été appliqué à SEDUITE actuellement en production. La conception itérative du modèle est répercutée et testée au fur et à mesure. Bien que la coordination de trois groupes de travail (un par système : SI , SU et SINT) ne soit pas simple, la difficulté la plus contraignante est le fait que SEDUITE soit utilisée.

En effet, les avancées technologiques en support aux SI, en particulier celles basées sur les Web Services rendent possibles un grand nombre d'adaptations dynamiques des services et des assemblages. La plate-forme Adore qui a servi de base pour la partie SI de notre expérimentation entre dans ce cadre. Nous trouvons les mêmes capacités au niveau des SINT. La plate-forme WComp qui a servi de base pour la partie SINT est particulièrement riche en terme d'adaptation, voire même d'auto-adaptation. Finalement, nous explicitons l'intervention de l'utilisateur à travers le SU. Il incombe alors au CDA de mettre en correspondance chaque systèmes.

Pour situer nos contributions, nous les positionnons par rapport à l'espace problème de la plasticité proposé dans (Calvary *et al.,* 2006). Celui-ci organise la plasticité selon plusieurs axes, à la base pour les IHMs, que nous étendons à tout le système d'information interactif :

**Remodelage : Modalité et Abstraction.** Avec notre modèle d'architecture, nous pouvons prendre en compte l'adaptation à tous les niveaux de l'Arche et grâce à l'approche Web Services pour Dispositifs et la réorganisation dynamique des as-

semblages des dispositifs, nous pouvons faire évoluer une modalité en n'importe quelle autre.

**Contexte d'usage : Environnement, Plate-Forme et Utilisateur.** Au niveau du Système Interactif, notre approche de l'adaptation porte sur le couple <plate-forme, environnement>. Au niveau du CDA, nous prenons en compte l'utilisateur par des préférences et nous visons la meilleure adéquation possible entre les services disponibles dans le SI avec ceux disponibles dans le SINT. Nous travaillons actuellement sur les algorithmes de composition d'IHMs abstraites, afin à terme d'automatiser au maximum les répercussions des évolutions.

**Méta-IHM.** A l'heure actuelle, le SU (notre méta-IHM) se limite à la gestion de préférence. Nous travaillons à la prévisualisation des conséquences de ces choix.

**Apprentissage.** C'est un point que nous n'avons pas abordé dans cet article. Mais nous l'envisageons par déduction des usages dans des contextes d'exécution différents. L'apprentissage peut alors se décliner selon deux axes : un calcul à la volée des orchestrations de services à appliquer ou par identification de "motif" par association contexte - solution. Ceci nécessite de constituer une base de motifs d'assemblage dès la conception et/ou par l'apprentissage.

**Redistribution.** Même si nous n'avons pas encore étudié la redistribution en terme d'ergonomie de la multimodalité, à l'instar des modalités, notre modèle permet la répartition des IHMs en sortie et en entrée par le CDA ou par le SINT, selon que l'utilisateur spécifie une préférence pour un type de dispositif ou selon l'apparition / disparition d'un dispositif.

**Mise en œuvre de la Plasticité.** L'approche orientée services permettant le remplacement dynamique et l'hétérogénéité des services, nous obtenons de facto une production des IHMs à la volée, le déploiement dynamique de l'adaptation et le mixage de plusieurs espaces technologiques. La granularité de l'adaptation est multiple : elle va du remplacement d'un service par un autre à une modification en profondeur des liens entre les services, l'usage et l'interaction. Le grain de reprise de l'usage du Système d'Information après adaptation est a priori au niveau de l'interaction, la maintenance de la cohérence de l'usage incombant au CDA. Finalement, la localisation des mécanismes d'adaptation est double : interne à chaque système et externalisée au niveau des Gestionnaires pour la propagation de l'évolution.

En résumé, nous avons posé les bases pour prendre en charge les adaptations de bout en bout, mais il reste à trouver les paramètres pour automatiser les adaptations, via une expression des utilisateurs.

## 5. Bibliographie

Annett J., Duncan K. D., « Task analysis and training design », *Journal of Occupational Psychology*, vol. 41, p. 211-221, 1967.

Balke W.-T., Wagner M., « Cooperative Discovery for User-Centered Web Service Provisioning. », *Proceedings of the International Conference on Web Services, ICWS'03, June 23 - 26, 2003, Las Vegas, Nevada, USA*, p. 191-197, 2003.

Ballagas R., Borchers J., Rohs M., Sheridan J. G., « The Smart Phone : A Ubiquitous Input Device », *IEEE Pervasive Computing*, January, 2006.

Bandara A., Payne T. R., de Roure D., Clemo G., « An Ontological Framework for Semantic Description of Devices », *3rd International Semantic Web Conference (ISWC)*, Hiroshima, Japan, November, 2004.

Bass L. J., Coutaz J., « A metamodel for the runtime architecture of an interactive system : the UIMS tool developers workshop », *SIGCHI Bull.*, vol. 24, n° 1, p. 32-37, 1992.

Blay-Fornarino M., Collet P., Lahire P., Lavirotte S., Pinna-Déry A.-M., Tigli J.-Y., Contrats et compositions de services de l'application SEDUITE (SErvices de Diffusion Ubiquitaire d'InformaTions dans des Etablissements scolaires), Technical Report n° F.4.1, RNTL Faros, January, 2007.

Calvary G., Coutaz J., Dâassi O., Ganneau V., Balme L., Demeure A., Sottet J.-S., « Métamorphose des IHM et Plasticité : Article de synthèse », *ErgoIA'06 proceedings*, 2006.

Calvary G., Demeure A., Coutaz J., Dâassi O., « Adaptation des interfaces homme-machine à leur contexte d'usage Plasticité des IHM », *RTSI Série ISI*, September, 2004.

Camara J., Canal C., Cubo J., Vallecillo A., « Formalizing WSBPEL Business Processes Using Process Algebra », *Foundations of Coordination Languages and Software Architectures (FOCLASA)*, Springer, San Francisco (CA), August, 2005.

Champalle O., David B., Chalon R., Masserey G., « Ordinateur porté support de réalité augmentée pour des activités de maintenance et dépannage », *in* , ACM (ed.), *3e Journées Francophone Mobilité et Ubiquité, Ubimob'06*, ACM Conference Proceedings Series, p. 103-106, sep, 2006.

Charfi A., Mezini M., « Aspect-Oriented Web Service Composition with AO4BPEL. », *ECOWS*, vol. 3250 of *LNCS*, Springer, p. 168-182, 2004.

Charfi A., Mezini M., « An aspect-based process container for BPEL », *AOMD '05 : Proceedings of the 1st workshop on Aspect oriented middleware development*, ACM Press, New York, NY, USA, p. 1-6, 2005a.

Charfi A., Mezini M., « Middleware services for web service compositions », *WWW '05 : Special interest tracks and posters of the 14th international conference on World Wide Web*, ACM Press, p. 1132-1133, May, 2005b.

Cheung-Foo-Wo D., Tigli J.-Y., Lavirotte S., Riveill M., « Contextual Adaptation for Ubiquitous Computing Systems using Components and Aspect of Assembly », *Applied Computing (IADIS)*, IADIS, Salamanca, Spain, February, 2007.

Christensen E., Curbera F., Meredith G., Weerawarana S., « Web Services Description Language (WSDL) specification », http ://www.w3.org/TR/wsdl, March, 2001.

Cohen J., Aggarwal S., « General Event Notification Architecture Base (GENA) », http ://quimby.gnus.org/internet-drafts/draft-cohen-gena-p-base-01.txt, July, 1998.

Courbis C., Finkelstein A., « Towards aspect weaving applications », *ICSE '05 : Proceedings of the 27th international conference on Software engineering*, ACM Press, New York, NY, USA, p. 69-77, 2005a.

Courbis C., Finkelstein A., « Weaving Aspects into Web Service Orchestrations. », *ICWS*, p. 219-226, 2005b.

Coutaz J., « PAC : an Implementation Model for Dialog Design », *Conference Proceedings of Interact'87*, Stuttgart, 1987.

Demeure A., Calvary G., Coutaz J., Vanderdonckt J., « The Comets Inspector : Towards Run Time Plasticity Control based on a Semantic Network », *Conference Proceedings of TAMODIA 2006*, 2006.

Douence R., « A Restricted Definition of AOP », *in* , K. Gybels, , S. Hanenberg, , S. Herrmann, , J. Wloka (eds), *European Interactive Workshop on Aspects in Software (EIWAS)*, September, 2004.

Dragicevic P., Fekete J.-D., « Étude d'une Boîte à Outils Multi-Dispositifs », *Conférence Francophone d'Interaction Homme-Machine (IHM)*, Cepadues, p. 55-62, September, 1999.

FIPA, « FIPA Device Ontology Specification », http ://www.fipa.org/specs/fipa00091, December, 2002.

Hariri A., Tabary D., Kolski C., « Démarche en vue de la Génération d'Interfaces Mobiles et Plastiques », *in* , ACM (ed.), *3e Journées Francophone Mobilité et Ubiquité, Ubimob'06*, ACM Conference Proceedings Series, September, 2006.

Hourdin V., Lavirotte S., Tigli J.-Y., Comparaison des systèmes de services pour dispositifs, Technical Report n° I3S/RR-2006-25-FR, Laboratoire I3S, Sophia Antipolis, France, August, 2006.

Jammes F., Smit H., « Service-oriented paradigms in industrial automation », *IEEE Transactions on Industrial Informatics*, vol. 1, n° 1, p. 62-70, February, 2005.

Joffroy C., Mosser S., Blay-Fornarino M., Plateforme ADORE : Aspect and Distributed ORchEstrations, Technical report, I3S, Sophia-Antipolis (France), March, 2007.

Keller A., Ludwig H., « The WSLA Framework : Specifying and Monitoring Service Level Agreements for Web Services. », *Journal of Network and Systems Management*, 2003.

Khalaf R., Mukhi N., Weerawarana S., « Service-Oriented Composition in BPEL4WS », *WWW (Alternate Paper Tracks)*, 2003.

Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C., Loingtier J.-M., Irwin J., « Aspect-Oriented Programming », *Proceedings of the European Conference on Object-Oriented Programming*, vol. 1241, p. 220-242, 1997.

Kuck J., Gnasa M., « Context-Sensitive Service Discovery Meets Information Retrieval », *Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, IEEE Computer Society, p. 601-605, 2007.

Lavirotte S., Lingrand D., Tigli J.-Y., « Définition du contexte et méthodes de sélection », *in* , J. Coutaz, , S. Lecomte (eds), *Secondes Journées Francophones : Mobilité et Ubiquité (UbiMob)*, IMAG, Grenoble, France, p. 9-12, June, 2005.

Le Meur A.-F., Balligand F., Blay-Fornarino M., Collet P., Rivierre N., Chapitre Service dans Livrable Faros : État de l'art sur la contractualisation et composition, Technical report, RNTL Faros, October, 2006.

MacKenzie M., Laskey K., McCabe F., Brown P., Metz R., Reference Model for Service Oriented Architecture 1.0, Technical Report n° wd-soa-rm-cd1, OASIS, February, 2006.

Martens A., « Analyzing Web Service based Business Processes », *International Conference on Fundamental Approaches to Software Engineering (FASE)*, vol. 3442, Springer-Verlag, Edinburgh (Scotland), April, 2005.

Martinez-Ruiz F. J., Arteaga J. M., Vanderdonckt J., Gonzalez-Calleros J. M., Mendoza R., « A first draft of a Model-driven Method for Designing Graphical User Interfaces of Rich Internet Applications », *LA-WEB '06 : Proceedings of the Fourth Latin American Web Congress (LA-WEB'06)*, IEEE Computer Society, Washington, DC, USA, p. 32-38, 2006.

Nemo C., Glatard T., Blay-Fornarino M., Montagnat J., « Merging overlapping orchestrations : an application to the Bronze Standard medical application », *International Conference on Services Computing (SCC 2007)*, IEEE Computer Engineering, Salt Lake City, Utah, USA, July, 2007.

OASIS, « OASIS UDDI Specification », http ://www.uddi.org/specification.html, July, 2002.

Paolucci M., Srinivasan N., Sycara K. P., Nishimura T., « Towards a Semantic Choreography of Web Services : From WSDL to DAML-S. », *Proceedings of the International Conference on Web Services, ICWS'03, June 23 - 26, 2003, Las Vegas, Nevada, USA*, p. 22-26, 2003.

Pauty J., Couderc P., Banâtre M., Synthèse des Méthodes de Programmation en Informatique Contextuelle, Technical Report n° 1595, IRISA, January, 2004.

Prehofer C., Bettstetter C., « Self-organization in communication networks : principles and design paradigms », *Communications Magazine, IEEE*, vol. 43, n° 7, p. 78-85, 2005.

Reenskaug T. M. H., « MVC XEROX PARC 1978–1979 », http ://heim.ifi.uio.no/ trygver/themes/mvc/mvc-index.html, 1979.

Schlimmer J., « Web Services Dynamic Discovery (WS-Discovery) », http ://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf, April, 2005.

Schlimmer J., Thelin J., « Devices Profile for Web Services », http ://schemas.xmlsoap.org/ws/2006/02/devprof/, February, 2006.

Sottet J.-S., Calvary G., Favre J.-M., « Towards Mapping and Model Transformation for Consistency of Plastic User Interfaces », Workshop CHI 06, 2006.

Tarpin-Bernard F., David B., « AMF : un modèle d'architecture multi-agent multi-facettes », *Techniques et Sciences Informatiques*, vol. 18, n° 5, p. 555-586, May, 1999.

Thévenin D., Adaptation en Interaction Homme–Machine : le cas de la plasiticité, PhD thesis, Université Joseph Fourier, Grenoble I, 2001.

Tigli J.-Y., Cheung-Foo-Wo D., Lavirotte S., Riveill M., « Adaptation au contexte par tissage d'aspects d'assemblage de composants déclenchés par des conditions contextuelles », *RTSI Série ISI*, vol. 11, n° 5, p. 89-114, 2006.

Ullmer B., Ishii H., « Emerging frameworks for tangible user interfaces », *IBM Systems Journal*, vol. 39, n° 3-4, p. 915-, 2000.

W3C, « Device Independence Activity », http ://www.w3.org/2001/di/, 2001.

W3C, « Semantic Web Services Ontology (SWSO) », http ://www.daml.org/services/swsf/1.0/swso/, May, 2005.

W3C, « Web Services Eventing (WS-Eventing) », http ://www.w3.org/Submission/WS-Eventing/, March, 2006.

Want R., Fishkin K. P., Gujar A., Harrison B. L., « Bridging Physical and Virtual Worlds with Electronic Tags », *SIGCHI conference on Human factors in computing systems : the CHI is the limit (CHI)*, Pittsburgh, Pennsylvania, USA, p. 370-377, 1999.

Weiser M., « The Computer for the Twenty-First Century », *Scientific American*, vol. 265, n° 3, p. 94-104, September, 1991.