
Un Système d'Agents à base de Composants pour les Environnements Domotiques

Fady Hamoui* — Marianne Huchard** — Christelle Urtado* —
Sylvain Vauttier*

¹ LGI2P / École des Mines d'Alès, F-30035 Nîmes cedex

{Fady.Hamoui, Christelle.Urtado, Sylvain.Vauttier}@mines-ales.fr

² LIRMM, UMR 5506, CNRS and Univ. Montpellier 2, F-34395 Montpellier Cedex 5
Huchard@lirmm.fr

RÉSUMÉ. Pour satisfaire les besoins des utilisateurs de manière pertinente et fiable, les systèmes de contrôle d'environnements domotiques doivent posséder des qualités de décentralisation, d'adaptation, de dynamique et d'autonomie. Nous proposons de répondre à ces exigences à l'aide d'un système multi-agents à base de composants. Les agents contrôlent les équipements de l'environnement et composent les services qu'ils fournissent afin d'exécuter des scénarios définis par les utilisateurs. Les architectures internes des agents sont dynamiquement modifiées pour adapter leurs comportements à l'ajout de scénarios. Cet article décrit le processus de mise en œuvre d'un scénario utilisateur, depuis la détection d'un équipement jusqu'au déploiement et à l'intégration de composants de coordination aux architectures des agents.

ABSTRACT. To meet user needs in a relevant and reliable way, home automation control systems must be decentralized, adaptable, dynamical and autonomous. We propose to answer to these requirements with a component-based multi-agent system. Agents control home devices and compose the services the devices provide to support user-defined scenarios. Agents' internal architecture is dynamically modified to adapt agents' behavior to scenario addition. This paper describes the process for user scenario support, from device detection to the deployment and integration of cooperation components in agents' internal architecture.

MOTS-CLÉS: Agent, Composants, Informatique Ubiquitaire, Architecture Orientée Service, Composition de Service, OSGi, UPnP

KEYWORDS: Agent, Components, Pervasive Computing, Service Oriented Architecture, Service Composition, OSGi, UPnP

1. Introduction

Les systèmes domotiques contrôlent des équipements disponibles dans les environnements domotiques, pour la réalisation de scénarios définis par les utilisateurs (*i.e.* faciliter les tâches du quotidien, gérer la sécurité, l'économie, l'écologie). Un environnement domotique est une habitation équipée d'un réseau d'équipements domestiques contrôlés par un système domotique. Chaque équipement fournit des services décrits par un ensemble d'opérations et émet des événements qui reflètent ses changements d'états. Le système domotique veille à l'exécution des scénarios définis sous la forme de scénarios. Un scénario implique la coordination de différents équipements de l'environnement. Considérons, par exemple, un environnement composé d'un volet, d'un radiateur et d'une horloge. L'horloge fournit un service qui offre deux opérations, pour modifier et obtenir l'heure. Ce service offre aussi un événement qui signale le changement de l'heure et informe de sa nouvelle valeur. Le scénario "température agréable le soir" se définit par "après 19H00, si la température du salon est inférieure à 17 ° C, le volet doit se fermer et le radiateur doit s'allumer à la puissance 6".

La plupart des systèmes domotiques actuels réalise des scénarios prédéfinis. Les utilisateurs du système n'ont pas la possibilité de définir eux-mêmes de nouveaux scénarios qui devraient être intégrés dynamiquement. Les systèmes domotiques doivent pourtant prendre en compte ces changements de contexte dans la réalisation de leurs scénarios. Comme il est irréaliste de disposer d'une bibliothèque couvrant l'ensemble des scénarios des utilisateurs et des configurations d'environnement possibles, une solution consiste à concevoir un système domotique capable de s'adapter aux différents environnements dans lesquels il est intégré. Ainsi, nous estimons qu'un système domotique doit posséder les qualités suivantes [BOT 07, BOU 06] :

Configurabilité. Il doit fournir aux utilisateurs des interfaces graphiques conviviales leur permettant de définir des scénarios dans leur environnement [BUR 01]. Ces interfaces doivent permettre l'utilisation de tous les services disponibles dans l'environnement auquel le système est intégré.

Décentralisation. Il doit être distribué pour être fiable, performant et tolérant aux pannes, en répartissant les scénarios sur plusieurs unités du système. Cela est cohérent avec le fait que les équipements sont distribués dans l'environnement.

Autonomie. Il doit limiter l'intervention des utilisateurs à la définition des scénarios et celle des administrateurs au paramétrage initial du système. La configuration du système pour s'adapter à l'ajout de scénarios doit être effectuée automatiquement et sans intervention humaine [CHE 06, GRO 06].

Dynamisme. Le système doit se configurer par rapport aux différents environnements et aux changements qui y ont lieu sans interrompre les scénarios en cours d'exécution et sans perturber le fonctionnement du système [CHE 06, GRO 06].

Dans cet article, nous présentons un système domotique géré par un système multi-agents à base de composants logiciels. La combinaison de ces deux paradigmes permet d'obtenir les qualités énoncées précédemment.

L'article est structuré de la manière suivante. Dans la section 2, nous présentons l'architecture de notre système domotique. La section 3 décrit le processus de définition et de réalisation d'un scénario. Dans la section 4, nous présentons les technologies utilisées pour l'implantation du système. Dans la section 5, nous comparons nos travaux aux systèmes existants. Enfin, la section 6 conclut et propose des perspectives.

2. Notre système domotique

Notre système est constitué d'un ensemble d'agents à base de composants. Les agents [BRI 01] sont des entités actives distribuées dans un environnement. Ils perçoivent des informations et réagissent aux événements qui surviennent dans leur environnement en exécutant des comportements. Ils interagissent selon un modèle de collaboration (échanges de services) pour la réalisation de scénarios complexes. Un composant logiciel [SZY 02] est une unité de programmation, autonome et découplée, qui peut être assemblée à d'autres composants au travers d'interfaces décrivant ses fonctionnalités et ses dépendances. Les assemblages de composants sont définis de manière explicite sous la forme d'une architecture décrivant comment les composants sont connectés entre eux. Cette architecture peut être dynamiquement modifiée par ajout, retrait ou substitution de composants ou par modification des connexions. Nous définissons la structure interne de nos agents à l'aide d'une telle architecture à base de composants afin de pouvoir adapter dynamiquement leur comportement. Nous définissons deux types d'agents, les agents interactifs (*Graphical User Interface Agent* ou GUIA) et les agents de contrôle des équipements (*Device Control Agent* ou DCA). Les GUIA sont responsables de l'interaction avec les utilisateurs et administrateurs du système. Ils sont installés dans des équipements qui possèdent des périphériques de saisie comme les ordinateurs ou les PDA. Ils offrent des interfaces graphiques permettant aux utilisateurs d'exécuter un service et de définir ou gérer les scénarios et aux administrateurs de paramétrer les agents DCA. Dans cet article, nous nous concentrons sur le mécanisme de réalisation des scénarios. Les DCA sont installés dans des boîtiers connectés à l'environnement, chaque boîtier est réservé à un DCA. Les DCA prennent en charge la détection, le contrôle des équipements et la réalisation des scénarios. Les équipements qu'ils peuvent contrôler sont spécifiés par les administrateurs. Les DCA contrôlent les équipements par la génération de composants de contrôle et réalisent les scénarios par la génération et l'assemblage de composants de coordination avec les composants de contrôle (adaptation dynamique de leur architecture interne).

2.1. Architecture globale du système domotique

La figure 1 représente l'architecture globale du système domotique. Lorsque le système est intégré dans un environnement, un processus d'initialisation est nécessaire pour qu'il soit opérationnel. Les GUIA lancent une opération de détection des agents présents dans l'environnement (1). De leur côté, les DCA détectent que les agents et les équipements (2). Les équipements fournissent des descripteurs qui permettent

4 Agents et composants pour la domotique

aux DCA, dans un premier temps, d'identifier leur type et les types de services qu'ils fournissent (3). Puis, chaque DCA envoie aux GUIA la liste des équipements et de leurs services qu'il a détectés (4). Les GUIA mettent à jour l'interface de configuration des DCA. Dans cette interface, chaque DCA est associé aux équipements qu'il a détectés (5). Le paramétrage, optionnel, d'un DCA consiste à spécifier la ou les catégories d'équipements et de services qu'il contrôle et la ou les pièces dont il a le contrôle de tous les équipements (pour réduire son champ d'action). Plusieurs DCA peuvent avoir le même paramétrage au même moment. Si un DCA n'est pas paramétré, il a un contrôle total des équipements de l'environnement (paramétrage par défaut) (6). Le paramétrage de chaque DCA est transmis à tous les GUIA pour qu'ils mettent à jour l'interface de configuration. Les DCA génèrent ensuite, en fonction de

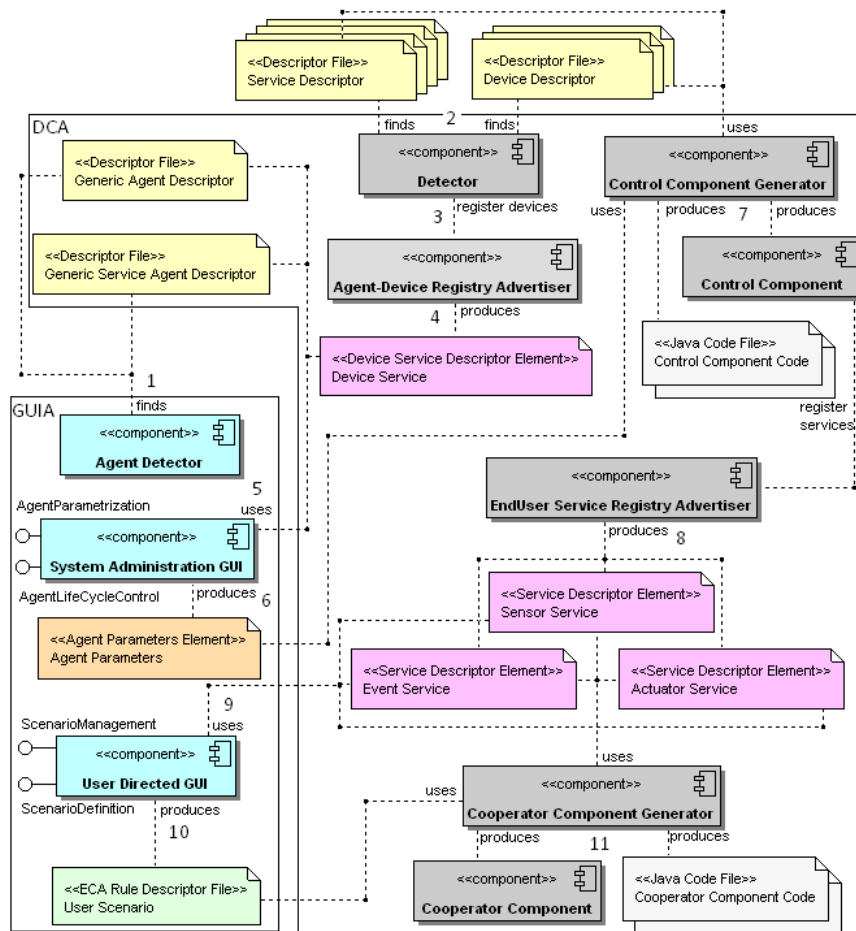


Figure 1. Architecture partielle pour la génération des composants

leur paramétrage, des composants pour contrôler les équipements (7). Par exemple, l'agent qui n'a accès qu'à la cuisine ou qu'à la télévision, ne génère que des composants de contrôle pour contrôler les équipements de la cuisine ou pour contrôler la télévision. Pour chaque équipement, un DCA extrait des descripteurs des services de l'équipement les signatures des opérations offertes et la liste des événements. Puis, il génère et déploie dynamiquement un composant de contrôle en se basant sur les descripteurs fournis par les équipements. Chaque service fourni par l'équipement est représenté par deux interfaces du composant : une de type actionneur qui contient les opérations du service qui exécutent des actions sur les équipements et l'autre de type capteur qui contient les opérations du service qui récupèrent les mesures fournies par les équipements. Les événements émis par les services de l'équipement sont fournis par une interface événementielle qui permet aux autres composants de s'abonner pour recevoir ces événements. Une fois les composants de contrôle générés, les DCA envoient aux GUIA des messages pour leur signaler quels services et quels événements ils contrôlent (8). Ces messages contiennent des descriptions orientées utilisateur des services fournis par les DCA. Enfin, les interfaces graphiques qui permettent aux utilisateurs de définir ou de sélectionner des scénarios réalisables sont mises à jour (9). Le système est alors opérationnel et les utilisateurs peuvent définir des scénarios. Chaque scénario est défini sous la forme d'une règle ECA (Event - Condition - Action) [JUN 07]. Le scénario est transmis sous forme d'un descripteur aux DCA participant à la réalisation du scénario (10). Lorsqu'un DCA reçoit le descripteur d'un scénario, il génère et déploie dynamiquement un composant de coordination pour la réalisation du scénario (11). Une fois généré, le composant se connecte aux composants de contrôle correspondant aux équipements impliqués dans le scénario. Chaque agent est constitué d'un assemblage initial de composants qui réalisent ses comportements de base (perception active de son environnement, sélection et exécution d'un comportement en réaction à un événement, contrôle de son cycle de vie, méta-contrôle de son architecture interne). Le modèle des composants de contrôle et de coordination est présenté dans [HAM 09]. Les architectures internes des GUIA et des DCA sont présentées dans les sections suivantes.

2.2. Les agents "Graphical User Interface Agent" (GUIA)

La figure 2 représente sous la forme d'un assemblage de composants la structure d'un GUIA. Les GUIA possèdent un composant principal "Core Component" (CC) qui est responsable de la communication avec les DCA. Les DCA sont automatiquement détectés par le composant "Agent Detector". Il les enregistre dans un annuaire géré par le composant "Agent Registry". Les descriptions des services orientés utilisateur fournis par les DCA sont enregistrés dans un annuaire de services géré par le composant *Operation-Event Registry*. Les interfaces interactives sont générées et mises à jour par le composant "End User Interface". Lorsque l'utilisateur définit un scénario, le composant "Scenario Descriptor Generator" génère un descripteur qui sera transmis aux DCA. Il enregistre les descripteurs dans un annuaire géré par le composant "Scenario Registry".

6 Agents et composants pour la domotique

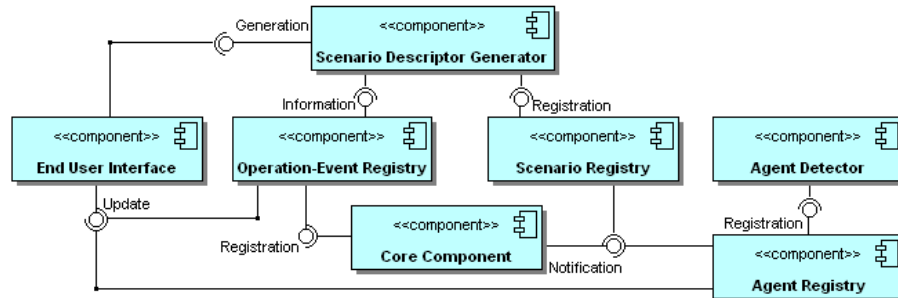


Figure 2. Structure interne des GUIA : Méta-niveau

2.3. Les agents "Device Control Agent" (DCA)

La figure 3 représente, sous la forme d'un assemblage de composants, la structure d'un DCA. À cette architecture viendront s'ajouter des composants de contrôle et des composants de coordination afin d'adapter le comportement de l'agent à un contexte d'utilisation spécifique. Les DCA possèdent un composant principal "Core Component" (CC) qui est responsable de la communication avec les autres agents. La détection des équipements présents dans l'environnement est effectuée par le composant "Device Detector". Il enregistre les agents dans un annuaire géré par le composant "Agent Registry" et les équipements dans un annuaire géré par le composant "Device Registry". Les descriptions de services orientées utilisateurs transmises au GUIA sont générées par le composant "End-User Service Registry". La génération des composants de contrôle et des composants de coordination est effectuée par le composant "Component Generator" qui les enregistre dans l'annuaire des composants géré par le composant "Component Registry".

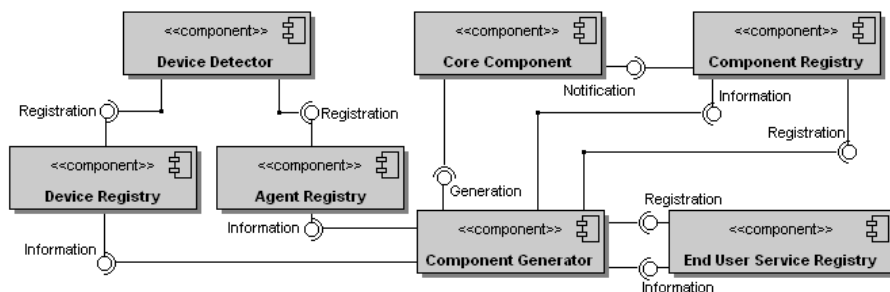


Figure 3. Structure interne des DCA : Méta-niveau

3. La définition et la réalisation d'un scénario

Cette section est basée sur l'environnement et le scénario décrit dans l'introduction. L'interface graphique qui permet de définir un scénario et le processus de définition d'un scénario sont détaillés dans [HAM 09]. Les scénarios sont définis sous la forme de règles ECA. La définition d'une règle combine un ensemble de services de types événement, capteur et actionneur fournis par les DCA dans ses clauses Event, Condition et Action. Le descripteur d'un scénario contient les informations techniques faisant le mapping entre la règle ECA définie par l'utilisateur et les services fournis par les DCA. Il se peut que plusieurs DCA fournissent le même événement ou les mêmes opérations. La stratégie de sélection des DCA qui participent à la réalisation du scénario vise à diminuer le nombre des messages échangés, en limitant le nombre des DCA participants et, notamment, en confiant la gestion de l'événement et de la condition à un seul DCA (celui qui contrôle le composant fournissant l'événement). L'utilité de cette stratégie est justifiée dans la section suivante. Une fois les DCA sélectionnés, le descripteur est envoyé à ces DCA.

Réalisation du scénario par un seul DCA. Le DCA génère et déploie dynamiquement un composant de coordination qui se connecte aux composants de contrôle qu'il requiert pour fonctionner (connexion des interfaces événement, capteur et actionneur). Une fois l'assemblage réalisé, le composant de coordination attend la notification de l'événement. Il vérifie alors la satisfaction de la condition et demande alors éventuellement l'exécution des actions. La figure 4 correspond à l'assemblage de composants qui réalise le scénario de notre exemple au sein d'un DCA.

Réalisation du scénario par plusieurs DCA. La figure 5 représente la réalisation par plusieurs DCA. Dans cette configuration, nous considérons que l'environnement domotique décrit précédemment possède quatre DCA paramétrés de telle façon que chaque DCA ne contrôle qu'un seul type d'équipement. Le DCAe, responsable de la vérification de l'événement, génère et déploie un composant de coordination qui

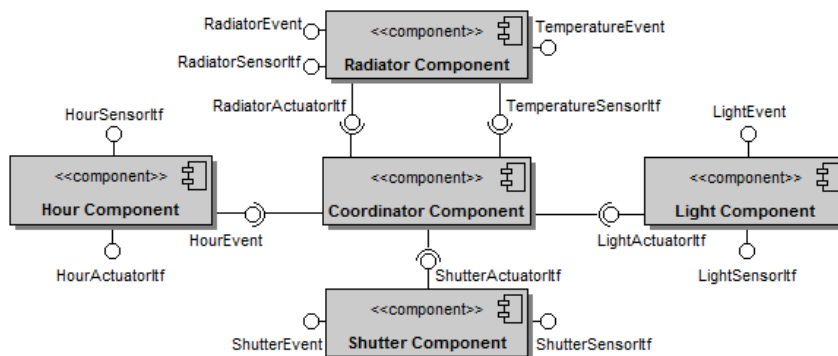


Figure 4. Assemblage de composants mettant en œuvre le scénario "température agréable le soir" dans l'architecture interne d'un DCA

se connecte au composant *Hour Component* qui lui notifie le changement de l'heure. Le DCAc, responsable de la vérification de la condition, génère et déploie un composant de coordination qui se connecte au composant *Radiator Component* qui lui fournit la température. La communication entre les agents est effectuée par le biais des composants *Core Component*. Lorsque l'événement est émis, le DCAe envoie un message (1) au DCAc qui vérifie la condition. Puis, lorsque la condition est satisfaite, le DCAc envoie un message (2) au DCAe l'informant de la satisfaction de la condition et des messages (3) de requête aux DCA participant à l'exécution des actions et une requête (4), en interne, au composant *Radiator Component*. Les DCAc, DCA1 et DCA2 exécutent l'allumage du radiateur, l'allumage de la lumière et la fermeture du volet, respectivement. Cependant, si entre-temps, l'événement n'est plus satisfait, donc avant la satisfaction de la condition, le DCAe informe le DCAc de cette situation. Le DCAc doit attendre à nouveau la satisfaction de l'événement.

4. Implémentation

Le méta-modèle du système est présenté dans [HAM 09]. Le système a été développé en Java en utilisant les technologies OSGi¹ et UPnP². OSGi est un environnement standardisé utilisé par l'industrie pour administrer dynamiquement l'installation

1. OSGi Alliance, <http://www.osgi.org>

2. UPnP forum, <http://www.upnp.org>

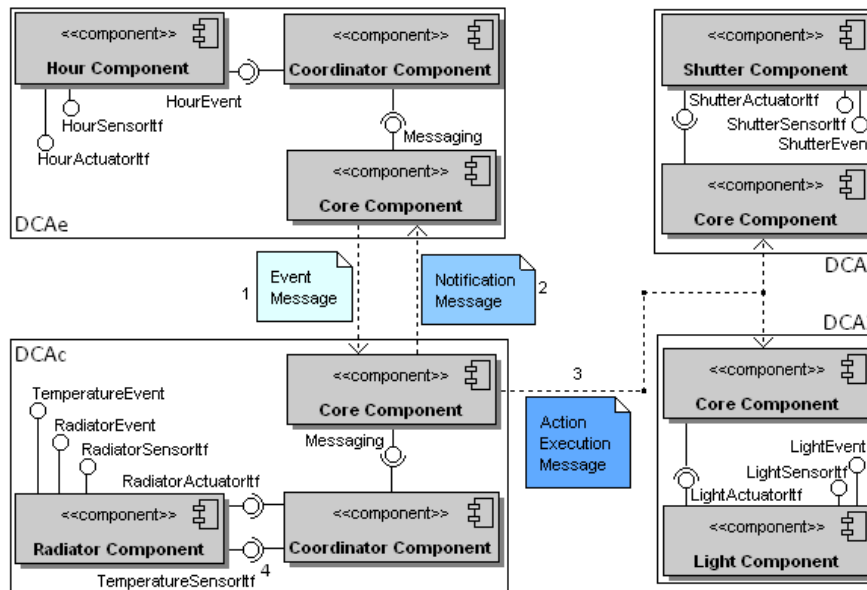


Figure 5. Structures internes des DCA qui mettent en œuvre le scénario "température agréable le soir"

de modules Java dans un environnement d'exécution. Ce standard propose un modèle de déploiement de "bundles" gérant leur interdépendance éventuelle grâce à des descripteurs définissant les services qu'ils requièrent et fournissent. Cette caractéristique nous permet d'utiliser les "bundles" OSGi comme une implantation des composants logiciels qui composent l'architecture interne des agents. Les composants logiciels peuvent être installés, mis à jour ou supprimés à l'exécution sans perturber le fonctionnement des autres composants. Les composants peuvent dynamiquement découvrir et utiliser d'autres composants grâce à un annuaire de services intégré à toute plateforme OSGi. L'annuaire permet aux composants de publier ou rechercher des services, et de s'abonner pour être informés de l'insertion, de la suppression ou de la mise à jour d'un service. Ces mécanismes sont utilisés pour construire automatiquement les assemblages de composants formant l'architecture interne des agents. Les classes des composants de contrôle et de coordination sont générées, compilées et encapsulées, à la volée, dans des bundles OSGi.

La technologie UPnP permet de créer un réseau d'équipements (serveurs) détectables et contrôlables par des Points de Contrôle (PC) (clients). Elle offre un mécanisme d'envoi de messages qui permet aux PC de détecter la présence, l'insertion, la suppression et la disparition d'un équipement du réseau, de contrôler les équipements et de recevoir des événements émis par les équipements. L'identification des équipements par les PC, pour les contrôler et recevoir leurs événements, est basée sur des descripteurs XML. Chaque équipement possède un ensemble de descripteurs qui contient les caractéristiques de l'équipement (marque, modèle, type, nom, ...), les caractéristiques des services qu'il offre (type, nom, ...), les URLs auxquels les PC doivent envoyer les commandes de contrôle des services et les demandes d'abonnement aux événements. Le descripteur d'un service contient l'ensemble des opérations, appelées "actions", qu'il offre et un ensemble de variables d'état, appelées "state variables". Le changement de la valeur d'une variable d'état est signalé aux PC abonnés par l'envoi d'un événement qui contient la nouvelle valeur de la variable. La technologie UPnP propose des descripteurs standardisés pour un ensemble de types d'équipements et de services³. Le composant principal de nos agents "Core Component" est implanté comme un équipement et un point de contrôle UPnP. Ainsi, les mécanismes de détection et de communication par messages ne sont pas uniquement utilisés entre les agents et les équipements mais entre les agents eux-mêmes. UPnP facilite la gestion du système multi-agents en fournissant un mécanisme de configuration automatique des ressources de communication des agents et de détection automatique de la présence d'autres agents. En tant qu'équipements UPnP, les agents possèdent un descripteur décrivant les services qu'ils fournissent aux autres équipements UPnP. Ils fournissent un service de communication qui permet aux agents de recevoir des messages de la part des autres agents, un service de contrôle du cycle de vie de l'agent qui permet aux administrateurs du système d'activer, de stopper ou d'éliminer l'agent et un service de paramétrage qui permet aux administrateurs du système de paramétrer, par le biais des GUIA, les équipements contrôlés par un DCA.

3. <http://www.upnp.org/standardizeddcps/default.asp>

Un prototype de notre système multi-agents a été réalisé et déployé sur un ensemble de conteneurs OSGi Felix⁴. Il met en œuvre un GUIA et un DCA contrôlant différents équipements virtuels (fournis par le projet CyberGarage⁵).

5. État de l'art

Nous classons les systèmes domotiques existants en trois catégories : Les systèmes centralisés, les systèmes basés sur des équipements intelligents et les systèmes basés sur les agents.

5.1. Les systèmes domotiques centralisés

[GRO 06] propose un système à base de composants pour l'assemblage dynamique et automatique de composants. L'assemblage nécessite quatre éléments : la description de l'application, un ensemble de composants, les données du contexte fournies par des capteurs et la politique qui dirige les décisions d'assemblage. La description d'une application consiste en un ensemble de configurations prédéfinies, assemblages de composants abstraits. Le système est responsable de sélectionner une configuration convenable et de choisir les composants correspondant aux composants abstraits.

[CHE 06] propose un outil qui permet aux programmeurs de définir des scénarios. Les scénarios sont décrits comme des aspects pouvant être tissés à l'assemblage de composants définissant l'architecture courante du système de contrôle pour y ajouter de nouveaux comportements. Le choix de l'application d'un aspect peut être déclenché par un utilisateur ou par la détection d'un équipement, permettant d'adapter le système aux changements de contexte.

[NAI 09] propose un système à base de composants qui supporte l'hétérogénéité des équipements. Chaque fabricant peut utiliser ce framework pour encapsuler ses technologies propriétaires dans des composants standardisés qui décrivent les fonctionnalités des équipements en termes de capteurs et d'actionneurs. Les capteurs et les actionneurs peuvent être combinés en scénarios consistant en une succession ordonnée d'actions élémentaires (l'écoute d'un capteur ou le déclenchement d'un actionneur).

[BER 03, NAK 06] proposent des systèmes qui détectent automatiquement les équipements dans leur environnement et génèrent des interfaces interactives qui listent les services fournis par les équipements détectés. Les utilisateurs interagissent avec les systèmes à travers les interfaces pour exécuter des services, mais ils n'ont pas la possibilité de définir des scénarios structurés et automatisés.

Outre leur architecture centralisée, les systèmes étudiés ne proposent pas, à l'exception de [CHE 06], de mécanismes permettant de composer dynamiquement différents

4. <http://felix.apache.org/site/index.html>

5. <http://www.cybergarage.org/>

scénarios. Cependant, [Che06] propose une approche par définition d'aspects qui s'adresse aux concepteurs des systèmes et non aux utilisateurs. Enfin, le tissage d'aspects nous semble moins flexible que l'évolution d'assemblages de composants pour gérer l'adaptation du comportement du système aux changements du contexte.

5.2. Les systèmes domotiques basés sur les équipements intelligents

Les systèmes suivants font l'hypothèse que tous les équipements possèdent des ressources suffisantes pour être programmés et interagir de manière autonome et directe avec les autres équipements (architecture pair à pair).

SodaPop [HEI 05] propose de documenter formellement les opérations des équipements à l'aide de règles décrivant leurs pré-conditions (état de l'environnement) et leur effets (changement d'états de l'environnement). Les besoins des utilisateurs sont exprimés comme les effets qu'ils souhaitent obtenir. Le système calcule automatiquement, par inférence de règles, des enchaînements d'opérations produisant les effets demandés par les utilisateurs.

[NAK 04] propose d'exprimer des scénarios sous forme de séquences d'exécutions de services, les paramètres de sortie d'un service pouvant être utilisés comme paramètres d'entrée du service suivant. L'ensemble des scénarios est découpé en fragments déployés sur chaque équipement sous forme de descripteurs. Chaque descripteur définit le comportement de l'équipement sous forme de pré/post conditions qui sont exécutées lors de l'invocation d'un service de l'équipement. Ces pré/post conditions consistent en des invocations d'autres services. Il est ainsi possible de définir des enchaînements d'exécutions de services requérant éventuellement des collaborations entre les équipements. La composition de scénarios multiples repose sur le concept de graphe d'intégration permettant le calcul d'un scénario global résultant, qui est découpé puis déployé sur l'ensemble des équipements. Ce calcul est centralisé et l'intégration d'un nouveau scénario entraîne la redéfinition complète du comportement de certains équipements.

Dans ces architectures décentralisées, les équipements possèdent un comportement autonome. Mais leurs ressources limitées ne permettent pas de les doter d'un comportement réflexif gérant leur adaptation dynamique aux changements de contexte. Seul [HEI 05] propose une telle solution, en intégrant aux équipements des mécanismes de raisonnement capables de calculer automatiquement des scénarios. Cette solution sophistiquée a des limites importantes. L'expression du besoin des utilisateurs est limitée par la liste des règles de comportement (effets) connues des équipements et l'ajout de nouvelles règles n'est accessible qu'à des utilisateurs "experts". Entre autres, la cohérence de l'ensemble des règles distribuées sur l'ensemble des équipements, la stabilité du calcul des scénarios sont des problèmes ouverts.

5.3. Les systèmes domotiques basés sur les agents

[WU 07] propose un système d'agents qui réalise les scénarios des utilisateurs par la génération d'agents mobiles. Chaque agent mobile correspond au scénario d'un uti-

lisateur. Pour réaliser un scénario, un agent mobile doit se déplacer d'un équipement à un autre en fonction de la localisation du service qu'il doit exécuter à chaque étape du scénario. Le scénario est défini comme une séquence de déplacements, d'invocations de services et d'exécutions de scripts (permettant de gérer des structures de contrôle). La définition du scénario exécuté par l'agent est statique, fixée au moment de sa génération. L'évolution des scénarios est reflétée par la génération d'agents mobiles ayant un nouveau comportement. Ainsi, [WU 07] remplace la distribution d'un scénario et la coordination entre agents par la mobilité d'un agent unique pour exécuter un scénario. Si l'agent mobile est détruit (planne de l'agent ou d'un équipement), tout le scénario est détruit. La performance du mécanisme de mobilité ne semble intéressante que dans des contextes très spécifiques. Chaque déplacement d'un agent est géré par la transmission d'un descripteur de l'agent à l'équipement qui doit instancier l'agent. Puis, cet agent doit interpréter le contenu du descripteur pour exécuter son comportement. Le gain du trafic réseau n'est obtenu que lorsque les mouvements des agents sont limités, donc pour des scénarios séquentiels (pas de "dialogue" entre équipements).

[SON 06] propose un système composé de systèmes multi-agents. Chaque système est associé à une pièce de l'environnement et contrôle les équipements de la pièce. Les événements émis par les équipements de l'environnement sont envoyés à tous les systèmes. Chaque système possède un entrepôt de composants (driver) pour le contrôle des différents types d'équipement et un entrepôt de règles associées à chaque composant de contrôle. Lorsque le système détecte un équipement, il déploie le composant de contrôle approprié et recherche les règles associées. Le système possède un composant responsable de la vérification des règles. Lorsqu'il reçoit un événement, il vérifie la satisfaction des règles, puis il demande l'exécution des services mentionnés dans les règles satisfaites. Un utilisateur peut se connecter à un système pour contrôler les équipements par l'invocation d'un service, mais il ne peut pas définir de scénarios personnalisés. Les capacités d'adaptation autonome d'un agent sont limitées par l'ensemble de ses composants de contrôle et l'ensemble, non redéfinissable par les utilisateurs, de ses règles de comportement.

Vallée *et al.* décrivent dans [VAL 06] un système multi-agents basé sur des mécanismes de composition dynamique de services basiques. Ils définissent trois types d'agents. Des agents d'assistance qui sont responsables du choix des scénarios à exécuter. Le choix est guidé par une commande (exécution d'un service) exprimée par l'utilisateur, ou bien guidé par des règles prédéfinies basées sur des informations du contexte. Des agents de composition qui sont responsables du choix des services qui participent à un scénario et de la détermination de la relation entre les services. Et des agents de service qui fournissent la liste des services disponibles dans l'environnement. L'architecture qui permet la composition des services, est décomposée en deux parties : une infrastructure de services et un système de composition. L'infrastructure de services gère la liste des services offerts par les équipements de l'environnement grâce aux descripteurs des services publiés par les équipements. Elle possède un système de gestion du contexte qui, grâce à des capteurs, collecte et maintient des informations sur le contexte. Les utilisateurs possèdent un "assistant personnel" qui

envoi des requêtes au système de composition pour satisfaire les besoins des utilisateurs. Le système de composition possède un générateur de plan qui fournit la description d'un scénario qui peut répondre à un besoin spécifique. Le mécanisme de composition de services utilise alors des informations sur les services et des informations sur le contexte pour sélectionner les services capables de réaliser chaque tâche du scénario. Ces informations sont formalisées par des concepts structurés en ontologies qui permettent le raisonnement pour le calcul automatique des scénarios.

On peut constater que les systèmes de contrôle multi-agents ont des niveaux de configurabilité très variables, allant de scénarios prédéfinis à des scénarios personnalisables, par la définition explicite de scénarios ou par le calcul automatique de scénarios répondant aux besoins des utilisateurs, exprimés de manière déclarative. À nouveau, cette solution semble complexe à mettre en œuvre si l'objectif est de prévoir le support de tous les besoins des utilisateurs. Il est probable que les utilisateurs expriment des besoins pour lesquels le système ne propose aucune réponse, sans solution simple permettant aux utilisateurs de configurer leur système.

6. Conclusion et perspectives

Nous avons développé un système d'agents, à base de composants, distribués dans un environnement domotique. Le système permet aux utilisateurs de définir leurs scénarios en fonction des services et événements fournis par les équipements de l'environnement. Les agents se connectent à l'environnement et se configurent en fonction du paramétrage des administrateurs par la génération de composants de contrôle. Des interfaces graphiques sont mises à jour pour permettre aux utilisateurs de définir leurs scénarios. Un scénario est défini sous la forme d'un modèle de coordination entre différents services. Il est réalisé par la génération de composants de coordination qui s'assemblent à des composants de contrôle et communiquent ensemble pour la réalisation du scénario.

Notre solution propose un moyen simple d'expression de scénarios utilisateurs, pour concentrer "l'intelligence" du système sur l'automatisation du déploiement et de l'évolution des architectures internes des agents, c'est-à-dire sur la composition de scénarios définis par l'utilisateur plutôt que sur la composition des scénarios eux-mêmes.

Des expérimentations ont été effectuées sur un petit ensemble d'agents et d'équipements. Nous envisageons de réaliser un environnement domotique virtuel contenant un plus grand nombre d'agents et une plus grande variété d'équipements. À ce niveau, nous devons gérer les conflits d'exécution entre les scénarios, la panne des agents ou des équipements (ce qui implique la réorganisation des DCA pour permettre la continuité des scénarios). En outre, la réalisation d'une hiérarchie de GUIA nous permettrait d'établir des droits d'accès aux services de l'environnement, par exemple, le GUIA des enfants n'aurait accès qu'aux équipements disponibles dans leurs chambres. Cela nous permettrait, aussi, d'introduire les GUIA mobiles inclus dans les PDA et

ordinateurs portables. Enfin, la réalisation de la coordination entre plusieurs environnements domotiques (immotique) autoriserait le partage de certains services pour la réalisation des scénarios et limiterait l'impact de la panne d'un équipement.

7. Bibliographie

- [BER 03] BERGER S., SCHULZRINNE H., SIDIROGLOU S., WU X., « Ubiquitous Computing in Home Networks », *IEEE Communications Magazine*, , 41(11) :128–135, Nov. 2003.
- [BOT 07] BOTTARO A., GERODOLLE A., LALANDA P., « Pervasive Service Composition in the Home Network », *IEEE 21st Int. Conf. on AINA*, Niagara Falls, Canada, pp 596–603, May 2007.
- [BOU 06] BOURCIER J., CHAZALET A., DESERTOT M., ESCOFFIER C., MARIN C., « A Dynamic-SOA Home Control Gateway », *IEEE Int. Conf. on SCC*, Chicago, USA, pp 463–470, Sept. 2006.
- [BRI 01] BRIOT J.-P., DEMAZEAU Y., *Principes et architecture des systèmes multi-agents*, Hermes/Lavoisier, 2001.
- [BUR 01] BURNETT M., CHEKKA S. K., PANDEY R., « FAR : An End-User Language to Support Cottage E-Services », *Proc. IEEE Int. Symp. on Human-Centric Computing Languages and Environments*, Stresa, Italy, pp 195–202, May 2001.
- [CHE 06] CHEUNG-FOO-WO D., TIGLI J.-Y., LAVIROTTE S., RIVEILL M., « Wcomp : a Multi-Design Approach for Prototyping Applications using Heterogeneous Resources », *IEEE Int. Wkshp on RSP*, Chania, Crete, pp 119–125, 2006.
- [GRO 06] GRONDIN G., BOURAQADI N., VERCOUET L., « MaDcAr : An Abstract Model for Dynamic and Automatic (Re-)Assembling of Component-Based Applications », *9th Int. Symp. on CBSE*, Västerås, Sweden, LNCS, 4063 :360–367, June 2006, Springer.
- [HAM 09] HAMOUI F., HUCHARD M., URTADO C., VAUTIER S., « Specification of a component-based domotic system to support user-defined scenarios », *Proc. of 21st Int. Conf. SEKE*, Boston, USA, July 2009, p. 597-602.
- [HEI 05] HEIDER T., KIRSTE T., « Smart Environments and Self-Organizing Appliance Ensembles », DAVIES N., KIRSTE T., SCHUMANN H., Eds., *Mobile Computing and Ambient Intelligence : The Challenge of Multimedia*, n° 05181 Dagstuhl Seminar Proceedings, IBFI, Schloss Dagstuhl, Germany, 2005.
- [JUN 07] JUNG J.-Y., PARK J., HAN S.-K., LEE K., « An ECA-based framework for decentralized coordination of ubiquitous web services », *Information & Soft. Tech.*, , 49(11-12) :1141–1161, Nov. 2007, Elsevier.
- [NAI 09] NAIN G., BARAIS O., FLEURQUIN R., JÉZÉQUEL J.-M., « EntiMid : un middleware aux services de la maison », *3^{ème} Conf. CAL'09*, Nancy, France, mars 2009.
- [NAK 04] NAKAMURA M., IGAKI H., TAMADA H., ICHI MATSUMOTO K., « Implementing integrated services of networked home appliances using service oriented architecture », *2nd Int. Conf. on SOC*, New York, USA, Nov. 2004, ACM, p. 269-278.
- [NAK 06] NAKAJIMA T., SATOH I., « A software infrastructure for supporting spontaneous and personalized interaction in home computing environments », *Personal Ubiquitous Comput.*, vol. 10, n° 6, 2006, p. 379–391, Springer-Verlag.

- [SON 06] SON M., SHIN D., SHIN D., « Design and Implementation of the Intelligent Multi-Agent System Based on Web Services », *IEEE Proc. of the 7th Inter. Conf. on WAIMW'06*, Washington, DC, USA, 2006, page 20.
- [SZY 02] SZYPERSKI C., *Component Software : Beyond Object-Oriented Programming*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [VAL 06] VALLÉE M., RAMPARANY F., VERCOUTER L., « Using Device Services and Flexible Composition in Ambient Communication Environments », *1st Inter. Wkshop on RSPSI*, Dublin, Ireland, May 2006.
- [WU 07] WU C.-L., LIAO C.-F., FU L.-C., « Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology », *IEEE Trans. on SMC, Part C*, vol. 37, n° 2, 2007, p. 193-205.