

Gestion intelligente d'un contexte domotique par un Système Multi-Agents

T. Dujardin^{*a}
tony.dujardin@lifl.fr

J. Rouillard^{*b}
jose.rouillard@lifl.fr

J-C. Routier^{*a}
jean-christophe.routier@lifl.fr

J-C. Tarby^{*b}
jean-claude.tarby@lifl.fr

^{*}Laboratoire d'Informatique Fondamentale de Lille, ^a Équipe SMAC, ^b Équipe NOCE, Université Lille 1 - Sciences et Technologies, France

Résumé

Dans le cadre de l'Intelligence Ambiante, cet article montre comment un système multi-agents prend en charge la gestion domotique d'une maison. Ce système intelligent peut gérer intégralement l'habitat ou se contenter d'effectuer les propositions d'action qu'il pense être les meilleures. Nous décrivons ici la conception et la mise en place d'un tel SMA basé sur l'approche orientée interactions. Ce travail constitue un exemple d'utilisation conjointe de technologies provenant à la fois de la communauté IHM et de la communauté SMA.

Mots-clés : Intelligence ambiante, Informatique Diffuse, Système mixte, Interaction, Communication, Simulation multi-agents

Abstract

In the context of Ambient Intelligence, this paper shows how a multi-agent system supports management of a home automation. This intelligent system can entirely manage your home or simply suggest any action it deems appropriate. We describe the design and implementation of a such MAS based on the interaction-oriented approach. This work is an example of joint use of technologies from both the HCI community and the MAS community.

Keywords: Ambient intelligence, Pervasive Computing, Mixed System, Interaction, Communication, Multi-agent Simulation

1 Introduction

L'intelligence ambiante est un paradigme dans lequel l'utilisateur interagit non plus avec un seul ordinateur, mais avec une multitude de machines, plus ou moins puissantes, reliées entre elles par un réseau. On parle alors d'une "informatique diffuse" ou de "pervasive computing" [13]. L'intelligence ambiante peut être appréhendée via quatre notions élémentaires :

1. L'ubiquité : la capacité pour l'utilisateur à interagir, de n'importe où, avec une multi-

tude d'appareils interconnectés via une architecture informatique distribuée.

2. L'attentivité : la faculté du système à recalculer en permanence le contexte d'usage, grâce à des capteurs disséminés sur le terrain.
3. L'interaction naturelle : la possibilité pour l'utilisateur de communiquer avec le système informatique le plus naturellement possible (parfois sans nécessité d'apprentissage) et de manière multimodale, en utilisant plusieurs de ses sens de manière alternée ou synergique [4].
4. L'intelligence du système : la faculté d'analyse du contexte et d'adaptation dynamique aux situations, d'apprentissage, de raisonnement (dans le sens d'un calcul probabiliste, par exemple), d'inférences et de déductions logiques.

Les systèmes multi-agents (SMA) sont étudiés, pour partie, dans cet objectif, et constituent un des moyens de contribuer à cet objectif d'intelligence ambiante. Dans le domaine des interactions homme-machine (IHM), l'étude des usages permet de rendre compte de l'utilité, de l'utilisabilité et plus globalement de l'ergonomie (efficacité, efficience, satisfaction...) des systèmes proposés.

Pour ce faire, nous avons collaboré au sein de deux équipes du laboratoire LIFL, afin de mettre en œuvre un système mixte capable de gérer intelligemment un contexte applicatif particulier (i.e. intelligence ambiante et domotique). Les chercheurs de l'équipe SMAC impliqués dans ce projet ont validé les aspects comportementaux et décisionnels (communauté Systèmes Multi-Agents) tandis que ceux de l'équipe NOCE ont validé les aspects interactionnels du système (communauté Interaction Homme-Machine).

Les attendus sont multiples. Il s'agit principalement :

- De la prise en charge effective par le SMA d'un contexte global d'interaction. Il est notifié d'événements et prend des décisions afin d'agir sur le monde, réel ou virtuel, en fonction d'objectifs déclarés par l'utilisateur ;
- De conseils et d'assistance aux utilisateurs sur la base de propositions d'actions. Le SMA calcule ce qu'il aurait fait s'il avait le contrôle total, mais ne fait que suggérer des actions auprès de l'utilisateur, qui, *in fine*, décide ou non de les appliquer ;
- De justification des choix de la part du SMA. Il devra, à terme, être capable d'expliquer ses prises de décisions ou plus précisément les critères qui ont conduit à ces décisions, comme par exemple, dans le cadre de la domotique, des critères énergétiques, de confort des utilisateurs, de sécurité, etc.

L'article s'articule de la manière suivante : la partie 2 présente un rapide état de l'art du domaine. Les parties 3 et 4 présentent respectivement le contexte expérimental et technologique de notre projet. La partie 5 est consacrée à la description du SMA utilisé. La partie 6 est relative aux expérimentations que nous avons mises en œuvre afin de valider notre approche, ainsi qu'aux extensions possibles. Enfin, nous concluons notre article avec la partie 7.

2 Contexte scientifique

Parmi tous les champs d'étude possibles de l'intelligence ambiante, celui du bâtiment intelligent nous paraît particulièrement intéressant, car il touche les utilisateurs dans leur quotidien. Les avancées dans ce domaine ont un impact sur une large population et sont applicables à différents niveaux : gestion de l'énergie, maintenance, nettoyage, sécurité, santé, autonomie des personnes âgées ou handicapées, vie de la famille, loisirs, activités professionnelles, logistique, etc. [5].

Historiquement, la domotique a été présentée, il y a une vingtaine d'années, comme devant révolutionner le quotidien des individus. Or, des facteurs bloquants, principalement dus aux coûts, à la fiabilité des solutions proposées, mais aussi à la difficulté d'adaptation d'un système générique à chaque habitation, ont conduit, sinon à un échec, tout au moins à un retard important dans ce domaine, pourtant porteur. Une infime partie de nos habitations possède des systèmes automatisés pouvant gérer des luminaires, des systèmes d'ouverture/fermeture de stores, portes, garages, voire de contrôle du chauffage et de la climatisation. Actuellement,

ces systèmes sont, pour la plupart, basés sur des mécanismes d'horloge. Quelques uns autorisent des manipulations à distance, par téléphone ou smartphone, et certaines solutions commerciales, comme le logiciel *HAL* (Home Automated Living), permettent un pilotage par commande vocale [1].

Beaucoup de projets explorent actuellement cette piste de recherche au confluent de la domotique et de l'intelligence ambiante. Le projet *Iroom* [3], par exemple, se fonde sur le besoin de populations hétérogènes d'utilisateurs à évoluer dans des environnements riches, et met en œuvre des solutions d'adaptation multimodales. Le projet ANR *Sweet Home* actuellement en cours au Laboratoire d'Informatique de Grenoble s'inscrit dans cette démarche d'accompagnement des personnes dans leur lieu de vie et s'appuie sur la notion de bâtiment sensoriel. Le programme *Adream* (LAAS, Toulouse) propose d'étudier conjointement le bâtiment intelligent optimisé en énergie (panneaux photovoltaïques notamment) et les objets communicants en vue d'une meilleure assistance aux personnes.

Des travaux autour du bâtiment intelligent et de la domotique faisant état d'utilisation de SMA apparaissent depuis de nombreuses années dans la littérature [12]. En France, ces travaux à base de SMA sont en partie orientés vers le contrôle et la gestion des énergies [2], [9]. Nos travaux vont également dans ce sens et il nous semble qu'une nouvelle étape sera franchie dans ce domaine dès lors que le bâtiment sera considéré comme une entité faisant partie intégrante d'un réseau communicant, et non plus comme un simple réceptacle. Le bâtiment demeure au cœur de ce dispositif car il possède la durée de vie la plus longue. Les autres composants devront, au fur et à mesure de leur apparition/disparition, se connecter de manière fluide, idéalement sans intervention humaine, et présenter aux utilisateurs, par introspection, leurs capacités à être utilisés de manière "naturelle".

3 Contexte expérimental

Afin de montrer comment une approche SMA permet la gestion effective d'une intelligence ambiante, dans le cadre de la domotique, nous nous sommes attachés à la mise en place effective d'un SMA. Son objectif est de piloter un certain nombre d'éléments domotiques au sein d'une pièce représentée sur la figure 1.

Cette pièce regroupe 4 bureaux disposant chacun d'une lampe n'éclairant qu'une zone limi-



FIG. 1 – La pièce à gérer : l’environnement réel.

tée de la pièce. La pièce dispose également d’un halogène capable de fournir un éclairage suffisant sur l’ensemble de la pièce. L’halogène consomme plus que chacune des lampes mais moins que les quatre lampes réunies. La pièce dispose également d’un radiateur et d’un climatiseur.

Dans ce contexte, les contraintes d’utilisation dont on souhaite disposer, et qui s’imposent donc au SMA à mettre en place, sont les suivantes :

- La température de la pièce doit être maintenue dans un intervalle de valeurs donné ;
- Lorsqu’un utilisateur approche d’un bureau, la lampe doit s’allumer si la luminosité est insuffisante. Un capteur de luminosité fournit cette information et des capteurs de présence situés sur chaque bureau permettent d’identifier la présence ou non d’une personne ;
- La gestion des différents éléments doit aller dans le sens des économies d’énergie :
 - Une lampe de bureau doit s’éteindre lorsque le bureau n’est plus occupé ;
 - Si la température de la pièce est trop élevée et si le radiateur est allumé, il convient d’éteindre le radiateur plutôt que de mettre en marche le climatiseur. Au moins dans un premier temps, en attendant que la température baisse avec l’arrêt du radiateur ;
 - Si les quatre lampes de bureau sont allumées, l’halogène doit être allumé et les lampes éteintes. On fournit ainsi assez de lumière aux quatre utilisateurs tout en diminuant l’énergie consommée.

Évidemment un humain peut également intervenir manuellement sur chacun des éléments, indépendamment de ces contraintes et ses interventions devront donc être transmises au SMA.

Enfin, en plus de la gestion des éléments évoqués et de la prise en compte des contraintes précisées, on souhaite que le SMA fonctionne

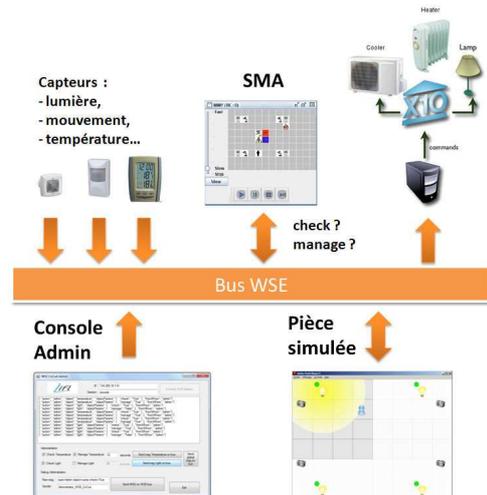


FIG. 2 – Présentation des communications entre le SMA, les périphériques et les interfaces de contrôle via le bus WSE.

selon deux modes. Le premier consiste à laisser le SMA gérer l’ensemble des dispositifs de la pièce (lampes, radiateur, etc.). Dans le second, il joue un rôle de conseiller en proposant l’action qui lui paraît la meilleure en fonction de la situation. Des modes hybrides peuvent également être envisagés : le SMA prenant par exemple en charge la gestion de la lumière et faisant des propositions pour la gestion de la température.

4 Contexte Domotique

4.1 Bus de communication WSE

Le système présenté ici s’articule autour d’un bus logiciel nommé WSE (Web Server Event) [11]. Tous les messages échangés par les différentes parties du système, par exemple les interactions utilisateurs ou les actions demandées aux périphériques, transitent par ce bus (cf. figure 2).

WSE est basé sur HTTP et nécessite seulement un serveur PHP pour fonctionner, par exemple EasyPHP ou LAMP. WSE est accessible actuellement en Javascript, C#, ActionScript 3 (Flex et Flash) et Java ; d’autres langages sont envisagés. Les deux seules contraintes pour qu’un langage puisse utiliser WSE sont de pouvoir traiter du JSON (JavaScript Object Notation) et de supporter les requêtes HTTP.

Les bénéfices de WSE sont d’être :

- **Multi-OS** : si un système d’exploitation peut accéder au web, il peut utiliser WSE. Par

conséquent, WSE est compatible avec Mac OS, Windows, Android, et Linux ;

- **Multi-plateformes** : le point précédent implique que WSE fonctionne aussi bien sur des ordinateurs que sur des Smartphones, des tablettes PC, etc. ;
- **Multi-navigateurs** : les deux points précédentes impliquent que WSE fonctionne avec des navigateurs comme Internet Explorer, Firefox, Chrome, Safari, Opera, du moment qu'ils supportent JavaScript ;
- **Multi-réseaux** : WSE est accessible par tous les réseaux qui permettent un accès web : réseaux filaires, mais aussi réseaux sans-fil comme le Wifi ou la 3G. Si les utilisateurs ont accès à Internet (c'est-à-dire que le port 80 est ouvert), alors ils ont accès au bus WSE qui, ainsi, n'est pas bloqué par les firewalls.

Les messages circulant sur le bus WSE sont des objets JSON. Ceci implique que chaque message doit respecter une structure JSON, par exemple `{"paramètre1" : "valeur1", "paramètre2" : {"paramètre2_1" : "valeur2_1"}}`. Chaque développeur peut créer son propre format de message (du moment qu'il respecte la structure JSON). Dans notre cas, nous avons utilisé le format suivant : `{"action" : "...", "actionParams" : "...", "object" : "...", "objectParams" : "...", "location" : "...", "locationParams" : "...", "fromWhere" : "...", "fromWhom" : "...}"` où chaque paramètre est optionnel et peut prendre les valeurs suivantes :

- `action` contient une action à exécuter par un périphérique ou pour le SMA. Dans le premier cas, on trouve des actions telles que `"switch on"` par exemple. Dans le cas du SMA, on trouve des actions comme : `"admin"` associée à `"check"` et `"manage"` (cf. section 5.3) et qui indique de quelle façon le SMA doit gérer le système domotique (prise en charge totale, partielle...) ; `"info"` qui correspond par exemple à des informations envoyées sur le bus par les capteurs ; et `"proposition"` qui indique que le SMA ne gère pas le système mais fournit des propositions d'actions ;
- `object` correspond à l'objet concerné (périphérique X10, caméra...) ;
- `location` indique l'endroit concerné par l'action et l'objet ;
- `fromWhom` précise qui a envoyé le message ;
- `fromWhere` indique d'où est parti le message ;
- Tous les `"...Params"` sont des paramètres pour le concept associé (pour l'objet, l'action...).

4.2 Domotique et X10

WSE permet la communication (réception d'informations, envoi d'ordres...) avec des périphériques réels. Il est par exemple possible de connecter des Phidgets (capteurs de lumière ou de température, potentiomètres, micro-moteurs, etc.) ou bien encore des périphériques domotique connectés par le protocole X10. Cet article présente, notamment, cette dernière solution.

Pour qu'un périphérique domotique X10 puisse recevoir une action à exécuter, il doit être relié au réseau électrique par l'intermédiaire d'un boîtier X10. Un boîtier contrôleur CM15 (un seul pour tous les boîtiers X10) est connecté à un PC, et transforme les ordres X10 du PC auquel il est relié en signaux électriques déclenchant les actions sur les boîtiers X10 correspondants. Dans le cas de notre expérimentation, nous avons utilisé un boîtier X10 pour contrôler une lampe et un autre pour contrôler un ventilateur (représentant le climatiseur).

5 Système Multi-Agents

La mise en place d'agents pouvant gérer un environnement réel soulève un certain nombre de problématiques.

- Nos agents évoluent dans un environnement simulé qui doit représenter au mieux l'environnement réel ;
- Afin de pouvoir contrôler les différents objets, le SMA doit pouvoir communiquer avec l'environnement réel en s'adaptant sur le protocole de communication de WSE (cf. figure 2) ;
- Le SMA doit soit gérer l'environnement, soit proposer des actions à faire par les humains dans leur environnement.

5.1 Approche centrée interaction

Pour assurer la gestion de l'environnement réel, nous utilisons l'approche centrée interactions pour des agents cognitifs situés [6]. Ces agents ont des buts qu'ils doivent résoudre en agissant dans l'environnement simulé. Ces actions se répercutent alors dans l'environnement réel.

Dans l'approche centrée interaction, un agent est défini par les interactions qu'il "peut-effectuer" (ses capacités) et qu'il "peut-subir". Une interaction est une action réifiable qui est exécutée par un acteur et subie par une cible. La méthodologie IODA [10] est utilisée pour définir les interactions attribuées aux agents.

Suivant l'affectation des interactions, nous distinguons trois types d'agents : actif, passif et artifice.

Actif : les agents actifs sont les agents capables d'effectuer des interactions. Ils possèdent un moteur comportemental qui est en charge du raisonnement et de son individualité.

Passif : les agents passifs sont les agents qui ne peuvent que subir des interactions.

Artifice : les artifiacts sont les agents qui ne subsistent, ni n'effectuent aucune interaction. Ils servent à faire des calculs ou comme entités contenant de l'information.

Propriétés dynamiques. Nos agents possèdent des propriétés. Une propriété est une association entre un nom et une valeur qui peut être de différentes natures (entier, flottant, booléen, etc.). Parmi ces propriétés certaines sont dynamiques tel que défini dans [7]. Une propriété dynamique est une propriété dont la valeur évolue durant la simulation indépendamment de l'exécution d'une interaction. On peut ainsi représenter des propriétés qui évoluent au cours du temps comme des propriétés homéostatiques (température interne, niveau d'énergie, etc.).

5.2 Agentification de l'environnement réel

Nous allons maintenant étudier comment mettre en place un SMA pour réaliser le contrôle domotique défini dans la partie 3.

Agents. Dans notre étude, nous proposons de contrôler un radiateur, un climatiseur, quatre lampes de bureau et une lampe halogène de l'environnement réel. Chaque objet contrôlable est représenté dans la simulation par un agent. Dans notre simulation, nous utilisons quatre types d'agents actifs : avatar, lampe, pièce et énergie. Les agents avatar représentent les humains. Les agents lampe représentent les lampes de bureau. L'agent pièce est chargé de la gestion de la température. L'agent énergie allume l'halogène si les lampes de bureau consomment trop d'énergie. Nous utilisons également trois types d'agents passifs : radiateur, climatiseur et halogène. Enfin, nous représentons les capteurs de présence par des artifiacts, type capteur.

Buts. Nous avons quatre types d'agents actifs : pièce, énergie, avatar et lampe. Chaque type d'agent possède un but spécifique définissant un état de l'environnement

agent	but
pièce	pièce.temp_min < pièce.data and pièce.temp_max > pièce.data
énergie	énergie.énergie_max > énergie.data
lampe	lamp.desk_sensor.detected = lamp.power
avatar	avatar.lumen < avatar.data

FIG. 3 – Les buts des agents actifs.

à satisfaire. Ces buts sont exprimés avec des expressions booléennes utilisant les propriétés des agents (cf. figure 3). Ainsi l'agent pièce vérifie que la température de la pièce (DATA) est comprise entre les bornes minimale (TEMP_MIN) et maximale (TEMP_MAX). L'agent énergie vérifie que la consommation énergétique courante (DATA) est en dessous du seuil maximal fixé (ENERGIE_MAX). Les agents lampe vérifient qu'ils sont éteints si leur capteur de présence ne détecte personne. Les agents avatar vérifient que leur niveau de luminosité perçue (DATA) est supérieur au seuil minimal fixé (LUMEN).

Propriétés. Tous les agents actifs de notre simulation possèdent une propriété MANAGE (cf. partie 5.3). Elle définit les types de messages envoyés (gestion ou proposition) par le SMA sur WSE. Lorsque cette propriété est à vrai, l'agent est en mode "gestion". Lorsque cette propriété est à faux, l'agent est en mode "proposition".

L'agent pièce, l'agent énergie et les agents avatar possèdent une propriété DATA. Celle-ci représente la donnée gérée par l'agent. Il s'agit de la température ambiante pour l'agent pièce, de la consommation énergétique de l'éclairage pour l'agent énergie et de la luminosité perçue pour les agents avatar.

Les agents radiateur et climatiseur possèdent une propriété NIVEAU qui représente le niveau de puissance du thermostat.

Les agents radiateur, climatiseur et lampe possèdent une propriété BONUS. Cette propriété représente le bonus gagné lorsqu'on allume ou le "bonus perdu" lorsqu'on éteint l'appareil. Ce bonus est une valeur positive pour l'agent radiateur (resp. pour les agents lampe), car elle représente une augmentation de la température (resp. de la luminosité perçue). Elle est négative pour l'agent climatiseur (une baisse de la température). La valeur de cette propriété est arbitraire. Elle est utilisée pour déclencher un chaînage dans la planification. Les valeurs réelles de luminosité et de température sont données par l'environnement.

```

nom :      allumer
condition : target.power==false
action :   target.power=true and
           add (actor.data, target.bonus)

```

FIG. 4 – Définition de l’interaction *allumer* par l’approche IODA

L’agent *lampe* possède la propriété *DESK_SENSOR*. Elle correspond au nom de l’artefact représentant le capteur de présence associé.

Les artefacts *capteur* possèdent la propriété *DETECTED* indiquant la présence ou non d’un humain dans leur champ d’action.

L’agent *pièce* possède les propriétés *TEMP_MIN* et *TEMP_MAX*. Elles représentent les bornes de température minimale et maximale à respecter.

Les agents *avatar* possèdent la propriété *LUMEN*. Cette propriété représente le niveau minimum de lumière que veut recevoir l’humain associé à l’avatar.

L’agent *énergie* possède la propriété *ENERGIE_MAX*. Elle représente le seuil d’énergie consommée par les lampes à partir duquel l’agent *énergie* doit éteindre les lampes de bureau pour allumer l’halogène.

Interactions. Les interactions du SMA sur l’environnement sont l’allumage et l’extinction des appareils électriques ainsi que l’augmentation et la diminution du niveau des appareils de chauffage. On retrouve donc quatre interactions : *allumer*, *éteindre*, *augmenter* et *diminuer*.

L’approche centrée interaction permet de définir chaque interaction une seule fois et de l’appliquer sur des cibles de différentes natures. Prenons par exemple, l’interaction *allumer* (cf. figure 4). Cette interaction peut s’effectuer sur une cible si elle est éteinte (condition). L’exécution de cette interaction a pour effet d’activer la cible et d’ajouter la valeur de la propriété *BONUS* de la cible à la propriété *DATA* de l’acteur. La matrice d’interaction présente les cibles sur lesquelles les interactions peuvent s’exécuter (cf. tableau 1). Par exemple, dans cette matrice nous voyons que l’agent *pièce* peut effectuer les interactions *allumer*, *éteindre*, *diminuer* et *augmenter* sur les agents *radiateur* et *climatiseur*, ce qui permet à l’agent *pièce* de gérer la température ambiante.

Prise de décision. Les actions exécutées par nos agents sont déterminées en deux phases. Pre-

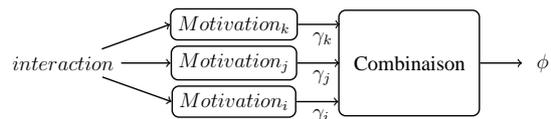


FIG. 5 – Fonctionnement de l’ASM basé sur les motivations.

mièrement, le **raisonnement** calcule les actions exécutables. Il s’agit des capacités de l’agent dont les conditions sont satisfaites. Deuxièmement, l’**individualité** de l’agent choisit l’action à exécuter.

Le **raisonnement** des agents actifs est délégué à un planificateur. Le raisonnement détermine donc, à partir des connaissances de l’agent, les séquences d’interactions, ou plans, à exécuter depuis l’état courant jusqu’à un état final satisfaisant ses buts. La première interaction d’une séquence est appelée interaction exécutable.

L’**individualité** est l’ensemble des facteurs propres à l’agent qui l’influencent dans le choix des interactions qu’il exécute. La gestion de l’individualité est déléguée à un mécanisme de sélection d’action (ASM pour “action selection mechanism”) basé sur des motivations [8]. Ce mécanisme choisit à chaque instant l’interaction que l’agent va exécuter parmi l’ensemble des interactions exécutables fournies par le raisonnement.

Pour être précis, l’ASM va évaluer chaque interaction exécutable en deux temps (cf. figure 5). Chaque motivation évalue chaque interaction exécutable (γ). L’ASM collecte et combine ces évaluations pour obtenir une évaluation globale (ϕ). Cette combinaison est nécessaire car les motivations sont définies indépendamment les unes des autres et sont en concurrence pour influencer le choix de l’agent. L’ASM sélectionne l’interaction qui a obtenu la meilleure évaluation. Comme décrit dans [7] les motivations peuvent être de différentes natures (environnementales, internes, etc.).

Motivations. Dans notre expérimentation, nous utilisons deux motivations pour l’ASM des agents actifs : l’*opportunisme* et les *préférences*. La motivation d’opportunisme favorise les interactions dont la distance entre l’acteur et la cible est courte. La motivation des préférences permet de personnaliser la gestion de l’environnement. Chaque agent actif associe une valeur de préférence à chaque interaction qu’il peut effectuer. Plus la valeur de la préférence est grande, plus l’agent sera motivé pour exécuter l’interaction.

Acteur \ Cible	\emptyset	radiateur	climatiseur	lampe	halogène
pièce		allumer, éteindre, diminuer, augmenter	allumer, éteindre, diminuer, augmenter		
énergie				éteindre	allumer, éteindre
avatar				allumer	
lampe	éteindre				

TAB. 1 – Matrice d’interactions des agents pour la simulation. Cette matrice définit les agents actifs (sur la première ligne) et les agents pouvant subir des interactions (sur la première colonne). La colonne \emptyset correspond aux interactions dont l’acteur est également la cible.

Pour cette simulation, nous utilisons la motivation des préférences de l’agent pour une gestion “écologique” (voir paragraphe 6.3). Ainsi, nous favorisons les interactions *éteindre* (valeur de préférence :10) et *diminuer* (valeur de préférence :20) au détriment des interactions *allumer* (valeur de préférence :1) et *augmenter* (valeur de préférence :2).

5.3 Communication entre le SMA et WSE

Tout agent (passif ou actif) qui représente un objet contrôlable est abonné aux messages envoyés sur WSE. Ces agents stockent dans une “boîte aux lettres” les messages les concernant et le cas échéant mettent à jour leurs propriétés et leurs connaissances. Ces mises à jour sont effectuées régulièrement de manière synchrone (le délai est fixé par l’utilisateur via la console d’administration et de manière indépendante pour la lumière et la température). Par exemple, si José se déplace dans l’environnement réel, l’agent nommé *avatarJosé*, mettra à jour ses coordonnées lors du traitement du message : `{"action" : "info", "object" : "human", "objectParams" : {"id" : "José", "x" : "6", "y" : "6"}, "location" : "LIFL" }`.

Comme nous l’avons vu, le système multi-agents s’exécute selon deux modes en fonction de la valeur de la propriété *MANAGE*. En mode **gestion**, le SMA envoie des messages sur le bus WSE afin de changer l’état d’un objet si besoin. En mode **proposition**, le SMA envoie des messages sur le bus WSE indiquant quelles actions il aurait effectuées. Ces modes sont choisis par l’utilisateur, indépendamment pour la lumière ou la température. La valeur de propriété *MANAGE* est mise à jour par l’envoi d’un message depuis la console d’administration. Le message envoyé pour le mode “gestion” de la température est : `{"action" : "admin", "actionParams" : {"manage" : "true"}, "object" : "temperature", "fromWhom" : "admin"}`. Le message envoyé pour le mode “proposition” de la lumière est : `{"action" : "admin", "actionParams" : {"manage" : "false"},`

`"object" : "light", "fromWhom" : "admin"}`.

Périodiquement, la console d’administration envoie à tous les agents actifs un message de déclenchement de la forme : `{"action" : "admin", "actionParams" : {"check" : "true"}, "object" : "OBJET", "fromWhom" : "admin"}` où *OBJET* vaut soit *temperature*, soit *light*. Les agents envoient alors en réponse un message, du type proposition ou gestion, qui dépend de leurs connaissances, capacités et buts. La périodicité d’envoi de ces messages peut être différente suivant l’élément à gérer (la température ou la lumière). Par exemple, un utilisateur peut fixer l’envoi de ces messages toutes les 30 secondes pour la gestion de la température et toutes les 3 secondes pour la gestion de la lumière. Ces messages sont de la forme : `{"action" : "switch on", "actionParams" : {"level" : "2"}, "object" : "heater", "fromWhom" : "SMA"}` pour l’allumage au niveau 2 du radiateur, ou de la forme : `{"action" : "proposition", "message" : "allumer sur lampeJosé", "location" : "LIFL", "fromWhom" : "SMA"}` pour une proposition d’allumage de la lampe qui se trouve sur le bureau de José.

6 Expérimentation

6.1 Les environnements : le réel et le simulé

Le système que nous avons développé (vidéo de démonstration disponible sur <http://www.lifl.fr/SMAC/projects/cocoa/domotique>) permet d’interagir avec un environnement simulé et/ou un environnement réel (le bureau avec les composants domotique X10). Pour l’environnement réel, comme indiqué précédemment, la connexion avec les périphériques réels dont nous disposons (une lampe et un ventilateur) passe par le protocole X10 grâce à un boîtier X10 relié à un PC lui-même connecté au bus WSE. Nous utilisons également des périphériques virtuels pour simuler des périphériques domotiques, que ce soit pour éviter d’utiliser les périphériques réels ou pour remplacer des pé-



FIG. 6 – Animation Flex-Flash représentant les capteurs de luminosité et de température, ainsi que les états d'une lampe de bureau, du radiateur et du climatiseur.

riphériques dont nous ne disposons pas (dans notre cas par exemple, le climatiseur est représenté par le ventilateur). Cette dernière solution permet également à quiconque de tester une configuration domotique avant même d'acheter et d'installer le matériel adéquat. Pour réaliser ces périphériques virtuels, nous avons utilisé la technologie Flex et Flash.

Flash permet de créer des animations (par exemple un ventilateur avec des pales qui tournent), alors que Flex se charge de la connexion au bus WSE. Ainsi, nous avons des pages Web contenant du Flex, lui-même contenant du Flash. Nous utilisons par exemple comme périphériques virtuels les capteurs de luminosité et de température (cf. figure 6). Grâce à ces deux capteurs, nous pouvons envoyer des informations au bus WSE comme le feraient de vrais capteurs (par exemple par SMP - Simple Messaging Protocol). Ce genre d'interface en Flex/Flash nous permet à la fois d'injecter facilement des données sur le bus (par exemple un changement de température dans la pièce) et de contrôler les informations qui circulent sur le bus (par exemple un ordre de mise en route du chauffage).

6.2 Les interfaces

Les interfaces de contrôle de notre système sont composées d'une IHM en C# représentant une console d'administration la console d'administration, d'une IHM en Flex/flash pour afficher la simulation de la pièce domotisée, et l'IHM du SMA (cf. figure 7) pour contrôler le bon déroulement de la simulation. Dans notre cas, la console d'administration (cf. figure 8) s'exécute sur un PC sous Windows 7 tandis que la si-



FIG. 7 – Environnement simulé vu par le SMA : les quatre bureaux avec les capteurs de présence et les lampes, au centre les agents halogène (H), pièce (P), radiateur (R), climatiseur (C) et l'agent énergie. Plus bas, se trouve un agent avatar.

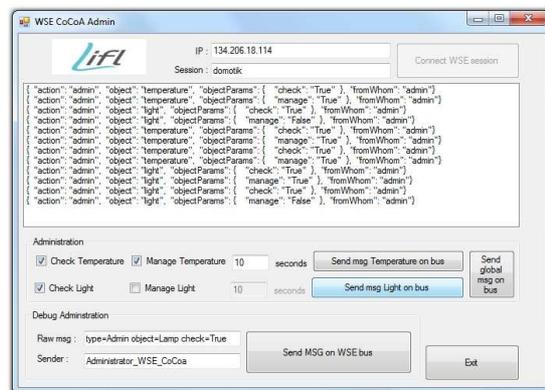


FIG. 8 – Console d'administration. Cette console permet à un utilisateur de choisir les modes de fonctionnement (i.e. proposition ou gestion) du SMA.

mulation de la pièce s'exécute sur un PC fonctionnant avec Windows XP. La troisième IHM s'exécute sous Linux.

L'interface de simulation de l'environnement permet de voir ce qui se passe quand une personne se déplace dans la pièce. Quand elle entre dans la zone de détection d'un capteur de mouvement, la zone correspondante est grisée et un message de détection de présence est envoyé sur le bus par le capteur. Dans certains cas (luminosité faible par exemple), la lampe située près de la personne s'allumera (cf. figure 9). Si la personne est à l'intersection de plusieurs zones de détection, chaque détecteur envoie un message et c'est le SMA qui prendra une décision (par exemple l'allumage d'une ou plusieurs lampes).

6.3 Observations et analyse

Le SMA se comporte comme attendu. Il parvient à prendre en compte les messages diffusés



FIG. 9 – L’avatar est détecté par deux capteurs (zones représentées en gris). Il choisit d’allumer la lampe la plus proche de lui par opportunisme.

depuis la console. Cela illustre que l’agentification de l’environnement réel a été correctement réalisée, le SMA arrivant dans le monde simulé à en reproduire le fonctionnement.

La gestion des différentes données (température, lumière) correspond au fonctionnement demandé : une température trop froide de la pièce entraîne le déclenchement du radiateur et un avatar qui s’approche d’une lampe provoque l’allumage de celle-ci. Elle s’éteint lorsque l’avatar s’éloigne. Ce comportement correct est dû au bon fonctionnement du moteur comportemental des agents, capable de résoudre les buts qui ont été fixés.

Dans le cas d’une température trop élevée avec un radiateur en fonctionnement, le système a la possibilité de diminuer le niveau de puissance du radiateur, de l’éteindre ou d’allumer le climatiseur. Le choix réalisé par le SMA est celui attendu : diminuer le radiateur. Ce choix s’explique par le fonctionnement du mécanisme de sélection d’action. On a fixé pour l’agent `pièce` une préférence plus importante à l’interaction `diminuer` qu’aux interactions `eteindre` et `allumer`. Cela a pour conséquence que les plans qui font intervenir l’interaction `diminuer` sont favorisés par rapport à ceux faisant intervenir `eteindre` ou `allumer`. Le mécanisme de sélection d’action proposant l’interaction exécutable dont le plan aura reçu la meilleure évaluation, c’est `diminuer` qui est sélectionnée.

Signalons que, si besoin, la diminution se poursuivra niveau par niveau. Cependant lorsque le niveau de puissance le plus bas est atteint, la diminution n’est plus possible. Le SMA choisira alors `eteindre` le radiateur plutôt qu’`allumer` le climatiseur pour une raison si-

miltaire : l’interaction `eteindre` est préférée à `allumer`.

Sur le même principe, lorsqu’un avatar s’approche suffisamment de deux lampes, dans un contexte où la luminosité est trop faible, le SMA n’allume que la lampe la plus proche de l’avatar (cf. figure 9). Cette fois c’est la motivation d’opportunisme qui joue un rôle. Cette motivation favorise d’autant plus une interaction que la distance entre l’acteur et la cible est faible. L’interaction avec la lampe la plus proche est donc la plus favorisée ; ce qui entraîne sa sélection.

6.4 Les extensions possibles

La justification des choix réalisés n’est pas encore explicitement exprimée par le SMA. Cependant nous disposons de toutes les informations issues des plans calculés par le moteur de raisonnement ainsi que des calculs réalisés par le mécanisme de sélection d’action. Le travail à accomplir pour que le SMA justifie les choix réalisés est donc relativement simple.

Dans un premier temps, on peut en effet sans aucune difficulté envisager que le SMA présente à l’aide des plans l’ensemble des actions possibles pour chaque élément. En justifiant à l’aide du chaînage arrière la pertinence de cette action par rapport à un but à satisfaire. Par exemple ainsi :

```
L'interaction proposée est
pièce.allumer(radiateur) car
(pièce.temp_min < pièce.data) est faux
or l'interaction allumer a pour conséquence
- radiateur.power = true
- add(pièce.data, radiateur.bonus)
et est exécutable car
- radiateur.power = false
```

Dans un second temps, il sera également possible de justifier le choix réalisé parmi les différentes actions possibles. Les calculs de la sélection d’action peuvent être détaillés pour chaque motivation et ainsi justifier le choix final. Par exemple dans le cas d’une température trop élevée avec un radiateur en fonctionnement, le système a la possibilité d’éteindre le radiateur, de le diminuer ou d’allumer le climatiseur. L’interprétation des valeurs obtenues dans chaque cas par la motivation des préférences permet d’expliquer le choix, même si dans ce cas le travail de mise en forme nécessaire se révèle certainement plus complexe, du moins pour être présentable de manière satisfaisante.

La mise en œuvre d’autres motivations est également envisageable. Par exemple, une motiva-

tion prenant en compte la consommation énergétique de chaque appareil. Pour cela il suffit d'associer pour chaque interaction et chaque cible une valeur correspondant à un coût énergétique. La motivation attribuerait une évaluation inversement proportionnelle à ce coût. Ainsi, la motivation favorisant les coûts les plus faibles privilégiera ouvrir les volets plutôt que allumer une lampe si on a besoin d'augmenter la luminosité ambiante.

7 Conclusion

Cet article illustre comment deux communautés (SMA et IHM en l'occurrence) peuvent collaborer au sein d'un même projet afin de faire évoluer chacune d'entre elles. Nous avons présenté ici la mise en œuvre effective d'un SMA pour le pilotage d'un système domotique. Le SMA dialogue avec le contexte domotique via un bus WSE. Les agents utilisés sont des agents cognitifs situés. La conception de SMA a été facilitée d'une part, par l'utilisation de l'approche centrée interaction qui a simplifié l'agentification des dispositifs domotiques à piloter et d'autre part, grâce à l'utilisation d'un mécanisme de sélection d'action basé sur les motivations pour la résolution des conflits de choix. Le SMA mis en place peut prendre en charge le contexte global d'interaction et agir sur l'environnement ou jouer un rôle de conseil et d'assistance aux utilisateurs par la suggestion des actions qu'il juge les mieux appropriées. L'expérimentation menée est tout à fait concluante et le SMA obtenu peut être assez facilement étendu par l'intégration d'autres dispositifs ou l'ajout d'autres règles de gestion. Cette première expérience permet d'envisager la mise en place d'un framework agent pour la conception de telles applications domotiques.

Références

- [1] Hal : Home automated living, <http://automatedliving.com>.
- [2] S. Abras, S. Ploix, S. Pesty, and M. Jacomino. Apport d'une approche multi-agents pour la résolution d'un problème de gestion de l'énergie dans l'habitat. In *Actes des JFSMA*, pages 141–144, 2009.
- [3] Y. Bellik, I. Rebaï, E. Machrouh, Y. Barzaj, C. Jacquet, G. Pruvost, and J-P. Sansonnet. Multimodal interaction within ambient environments : An exploratory study. In *Proceedings of INTERACT'09*, pages 89–92, 2009.
- [4] Y. Bellik and D. Teil. Définitions terminologiques pour la communication multimodale. In *actes des 4ièmes Journées sur l'ingénierie des interfaces Homme-Machine*, pages 229–232. Telecom Paris Publ., 1992.
- [5] J. Coutaz and J. Crowley. Plan intelligence ambiante : Défis et Opportunités. Document de réflexion conjoint du comité d'experts Informatique Ambiante du département ST2I du CNRS et du GT Intelligence Ambiante du GCS3 du MESR, DGRI A3, 2008.
- [6] D. Devigne, P. Mathieu, and J-C. Routier. *Intelligence Artificielle et Jeux*, chapter Simulation de comportements centrée interactions, pages 183–210. Hermès, 2007.
- [7] T. Dujardin. *Construction d'individualité par mécanisme de sélection d'action basé sur les motivations*. PhD thesis, Université Lille 1, Novembre 2010.
- [8] T. Dujardin and J-C. Routier. Behavior design of game character's agent. In *Proceedings of IAT'10*, pages 423–430, 2010.
- [9] J. Gil-Quijano, C. Herpson, and N. Sabouret. Prédiction de l'activité humaine afin de réduire la consommation électrique de l'habitat. In *Actes des JFSMA*, 2010.
- [10] Y. Kubera, P. Mathieu, and S. Picault. Interaction-oriented agent simulations : From theory to implementation. In *Proceedings of ECAI'08*, pages 383–387, 2008.
- [11] X. Le Pallec, R. Marvie, J. Rouillard, and J-C. Tarby. A support to multi-devices web application. In *23rd symposium on UIST '10*, volume 4946 of *LNAI*, pages 391–392, New-York, USA, 2010.
- [12] G. Clarke S. Sharples, V. Callaghan. A multi-agent architecture for intelligent building sensing and control. *International Sensor Review Journal*, 19(2) :135–140, may 1999.
- [13] M. Weiser. The computer for the 21st century. *Scientific American*, 65(3) :66–75, 1991.