

Aide à la conception d'un système domotique pour l'assistance aux personnes à mobilité réduite

Willy Allègre - Thomas Burger - Pascal Berruet

Université de Bretagne-Sud
Lab-STICC - Centre de recherche
Rue de Saint Maudé - BP 92116
56321 Lorient Cedex, France.

Contact : `prenom.nom@univ-ubs.fr`

Résumé

Les personnes à mobilité réduite éprouvent des difficultés importantes, sinon une incapacité physique totale à effectuer les tâches quotidiennes de manière autonome. Une des solutions permettant de compenser ces incapacités motrices consiste à s'appuyer sur des technologies d'automatisation du fonctionnement de l'habitat. De nombreuses initiatives témoignent de la dynamique autour des habitats intelligents, de la domotique et de l'aide aux personnes dépendantes. Cependant, très peu de ces réalisations se sont intéressées à la conception de systèmes intelligents pour permettre aux personnes non-expertes de ces technologies de modéliser un environnement domotique et de l'adapter aux personnes à mobilité réduite. Ce travail de conception revient actuellement à un technicien plutôt qu'à un ergothérapeute ou toute autre personne capable de prendre en compte les besoins de vie exprimés par ces personnes. Pour répondre à ces problématiques, cet article propose un flot de conception comprenant une approche composant pour modéliser l'architecture du système et une offre de services répondant aussi bien aux besoins des concepteurs que des utilisateurs. Basé sur l'Ingénierie Dirigée par les Modèles, il permet d'automatiser la génération du code de commande pour les systèmes domotiques.

Mots-clés : Domotique, Assistances techniques, Habitats intelligents, IDM, Outils et techniques de conception

Abstract

People with limited mobility have considerable difficulties or even physical incapacities to perform daily tasks independently. The use of home automation is one way to overcome these motor disabilities. Many works have shown the dynamics of smart homes, home automation and assistance for dependents. However, very few of these accomplishments have focused on the design of intelligent systems which would allow non-experts to model and to adapt a home automation environment for people with reduced mobility. This design work is currently restricted to technicians, rather than occupational therapists or others who are able to best understand the needs of those with limited mobility. To take up these challenges, this paper proposes a design flow including a component approach for modeling system architecture and a range of services to meet the needs of both developers and users. Based on Model Driven Engineering, it automates the control code generation for home automation systems.

Keywords: Home automation, Assistive technologies, Smart homes, MDE, Design Tools and Techniques

1. Introduction & Contexte

Les personnes à mobilité réduite présentent de réelles difficultés, voire une incapacité physique totale à effectuer les tâches quotidiennes de manière autonome. En effet, qu'elles soient des personnes handicapées moteur (paraplégiques, tétraplégiques) ou qu'elles fassent partie de la population vieillissante, elles présentent des capacités motrices réduites ne permettant pas ou difficilement de commander la multitude d'équipements présents dans leur habitat. La nouvelle définition du handicap [16] introduit la notion de contexte et d'environnement de vie de la personne. Dès lors, la domotique¹ est une solution envisageable pour compenser le handicap des personnes à mobilité réduite, en leur permettant de réaliser les activités de la vie quotidienne sans l'aide systématique d'une tierce personne. Pour se faire, le système doit être capable de faciliter voire d'automatiser l'activation de certains services et de proposer aux utilisateurs des services adaptés. Cela implique un niveau d'interaction élevé donnant l'illusion d'un habitat intelligent.

La domotique intelligente rejoint le domaine de l'informatique ubiquitaire [17]. Un système ubiquitaire est composé d'objets communicants capables de traiter de l'information et de proposer des services de manière indépendante. (Les termes d'*informatique pénétrant* ou d'*intelligence ambiante* sont aussi utilisés.) Du fait du nombre important d'équipements « intelligents » potentiellement présents dans un environnement domotique, il devient complexe pour les utilisateurs d'assurer le contrôle et le suivi de ces objets. Différents travaux se sont intéressés à la composition flexible et automatique de services de haut niveau dans le but de faciliter la vie de l'utilisateur. L'architecture logicielle est bien souvent basée sur un Système Multi-Agents (SMA) prenant en compte de manière dynamique les besoins des utilisateurs ainsi que des informations sur le contexte, pour adapter le processus de composition de services [9]. D'autres travaux ont vu le jour pour la composition de services de haut niveau : on parle « d'applications dirigées par les buts » [11]. Les besoins exprimés par l'utilisateur sont traduits par un objectif de haut niveau que le système est chargé d'atteindre.

Ces initiatives sont récentes et témoignent d'une certaine dynamique autour de la thématique des *habitats intelligents* pour l'assistance aux personnes dépendantes. Néanmoins, très peu de ces réalisations [10] se sont intéressées à la conception de systèmes intelligents pour permettre aux personnes non-expertes de ces technologies de modéliser, de concevoir et d'adapter selon leurs souhaits un environnement domotique. Il semble pourtant naturel que l'adaptation d'un lieu de vie pour une personne en fauteuil roulant soit faite par un ergothérapeute, plutôt que par un domoticien. De plus, les quelques contributions apportées pour l'aide à la conception de tels systèmes sont bien souvent limitées du fait de la modélisation adoptée [2]. Ils ne sont par exemple pas facilement généralisables à un espace de vie important où cohabitent plusieurs personnes (ex : passage d'un modèle de chambre individuel à un modèle de centre de soins complet).

Cet article présente une approche de modélisation pour l'aide à la conception de systèmes domotiques d'assistance. Dans la section 2, nous détaillons les points de vue du concepteur (non-technicien) et de l'utilisateur d'un tel habitat, afin de clairement définir les verrous scientifiques auxquels nous devons faire face. Ensuite, dans la section 3, nous proposons une méthode de modélisation pour les systèmes domotiques qui est adaptée au point de vue du concepteur. Cette méthode fait appel à l'Ingénierie Dirigée par les Modèles (IDM) et adopte une approche composant pour la construction d'un environnement domotique. Afin de tenir aussi compte du point de vue de l'utilisateur dépendant, nous proposons dans la section 4 de coupler cette modélisation architecturale à une gestion intelligente des équipements, basée sur la définition de *services* et de *modes* adaptés. Enfin, dans la section 5, un flot de conception illustré permet de valider de manière informelle l'approche de conception retenue dans nos travaux.

1. La domotique regroupe l'ensemble des technologies de l'électronique, de l'informatique et des télécommunications permettant le contrôle (via une interface homme-machine) et l'automatisation des équipements de l'habitat.

2. Positionnement

Deux acteurs peuvent être identifiés dans nos travaux : il s'agit du concepteur et de l'utilisateur. Le *concepteur* n'est pas un technicien expert en automatique, mais une personne sensibilisée à la domotique, chargée avant tout de construire un environnement de vie adapté à l'*utilisateur* qui présente des capacités motrices réduites. Ainsi, le concepteur est avant tout quelqu'un capable de prendre en compte cette situation de handicap (un ergothérapeute par exemple).

Du point de vue de l'utilisateur, des interactions de haut niveau avec le système sont indispensables pour deux raisons. Premièrement, elles permettent d'alléger la charge cognitive² de l'utilisateur [6] en lui proposant des interactions intuitives : *Sortir* présente un niveau sémantique supérieur à *Ouvrir la porte*. Mais ces interactions permettent surtout de compenser son handicap physique en facilitant ou en automatisant l'activation de services [12] : le service composé *Sortir* permet d'activer plusieurs services élémentaires comme *Ouvrir la porte*, *Eteindre les lumières*, etc.

De plus, un système domotique doit être facile à concevoir, facile à maintenir, facile à commander et flexible. Tous ces points rendent l'utilisation des outils de CAO (Conception Assistée par Ordinateur) indispensable pour le concepteur. Paradoxalement, il existe peu d'outils et de méthodes de conception de haut niveau adaptés à ces systèmes complexes. Ils sont actuellement développés en utilisant des procédures de bas niveau et n'ont pas de méthodologies qui permettent une modélisation indépendante d'une plateforme technologique [8]. Les solutions commerciales [1,2] sont dépendantes d'une technologie et bien souvent la syntaxe qu'elles mettent à disposition des concepteurs est restreinte et peu intuitive. Les concepteurs passent dans ce cas plus de temps à prendre en main, à s'approprier la syntaxe du logiciel qu'à s'occuper réellement de la partie métier de la conception. De ce fait, l'installation de tels systèmes est laissée aux experts du domaine. Ces derniers sont chargés, une fois les besoins de l'utilisateur recueillis, de déployer les équipements domotiques et de les programmer, bien souvent à travers différents outils logiciels propriétaires.

L'objectif de ce travail est donc double. Non seulement, nous cherchons à proposer une méthode de conception de la commande d'un système domotique qui soit accessible à une personne non-experte, mais également que le système ainsi conçu puisse fournir des interactions de haut niveau sémantique pour satisfaire les besoins des personnes à mobilité réduite.

3. Modélisation pour la conception d'un système domotique

Dans cette section, nous nous intéressons au point de vue du concepteur chargé de construire un environnement domotique. Nous proposons dans un premier temps d'appliquer aux systèmes domotiques les concepts issus de l'IDM pour rendre accessible leur conception à des personnes non-expertes de ce domaine. Dans un second temps, une modélisation ascendante par agrégation de composants est proposée pour rendre cette conception intuitive. Nous présentons ainsi les *points de vue* ainsi que les *concepts* définis dans le modèle retenu.

3.1. L'Ingénierie Dirigée par les Modèles (IDM)

L'IDM se distingue des différentes approches issues du génie logiciel. Sa démarche pose comme principe que tout ou partie d'une application informatique est généré à partir de modèles. L'architecture MDA³ fournit des outils, concepts et langages pour créer et transformer des modèles, développant ainsi une communauté importante autour de ce domaine.

Cette approche est basée sur trois concepts fondamentaux : le modèle, le métamodèle et la transformation de modèles. Un modèle est une représentation partielle d'un système physique qui peut être défini via un langage métier de description textuel ou graphique appelé DSL (Domain Specific Language). Le métamodèle définit la structure du langage (grammaire) utilisée pour construire ce dernier (tout modèle est conforme à un métamodèle). Enfin, la transformation de modèles définit des règles pour traduire un modèle source en un ou plusieurs modèles cibles.

2. La charge cognitive peut devenir excessive du fait de la gestion simultanée de plusieurs équipements et du manque d'automatisation (charge intrinsèque) ou de la manière dont sont présentées les informations (charge extrinsèque).

3. MDA (Model Driven Architecture) : <http://www.omg.org/mda/>

La manipulation d'un modèle propre à un domaine permet au concepteur de se focaliser sur les fonctionnalités plutôt que sur l'implémentation réelle du système. En augmentant ainsi le niveau d'abstraction, la conception devient intuitive et de ce fait plus accessible aux personnes non-expertes des technologies ciblées. L'implémentation est alors automatisée à travers l'utilisation de transformations de modèles. À partir d'un modèle indépendant de toute plateforme (PIM⁴) utilisant un langage approprié (ex : DSL), elles permettent de générer un ou plusieurs modèles spécifiques à une plateforme (PSM⁵) afin de générer le code fonctionnel du système.

Nous pensons que l'utilisation conjointe d'un DSL et d'une approche dirigée par les modèles peut offrir un niveau d'abstraction suffisant pour rendre la conception de systèmes domotiques d'assistance accessible à une personne non-experte. Tout d'abord, la génération automatique du code de commande des différents équipements domotiques constitue un point essentiel dans le processus de conception. Elle doit faire face à la diversité des protocoles et moyens de communication (KNX⁶, bluetooth, infrarouge...) associés aux équipements présents dans l'habitat. Les transformations de modèles telles que nous les utilisons y apportent une solution : à partir d'un modèle métier indépendant de toute plateforme (PIM), elles permettent de générer des modèles spécifiques à une plateforme (PSM) à partir desquels le code de commande spécifique à un protocole est généré. Nous proposons également de simplifier la conception d'un système domotique par une modélisation de haut niveau sémantique afin de faciliter le travail du concepteur, mais aussi afin de limiter son besoin de compétences techniques. La possibilité de manipuler un modèle dédié aux environnements domotiques permet au concepteur de s'abstraire de toute contrainte liée à la manipulation d'un langage de programmation spécifique. Il peut ainsi se concentrer à la fois sur les besoins de modélisation de l'espace de vie et sur les besoins du potentiel utilisateur. Une telle modélisation est rendue possible par l'utilisation d'un langage de description orienté métier (DSL), qu'il soit textuel ou graphique. Les auteurs de [8] proposent un langage graphique dédié à la conception de systèmes domotiques en utilisant l'ingénierie dirigée par les modèles. L'utilisation d'un tel langage permet de faciliter la phase de sélection/spécification de manière visuelle et intuitive pour la représentation de tels systèmes.

3.2. Approche composant pour la modélisation d'un système domotique

L'approche composant telle qu'elle est définie dans le paradigme CBSE [5] (Component Based Software Engineering) définit qu'un composant est une « boîte noire » ne présentant que ses interfaces et ses exigences. Un composant est considéré comme *substituable*, *réutilisable* et *composable* (des composants peuvent être couplés pour obtenir des fonctions plus complexes).

Des travaux issus des SAP (Systèmes Automatisés de Production) [13] mettent l'accent sur l'approche ascendante de modélisation par agrégation de composants pour les systèmes transitiqes. Dans ces travaux, un composant logiciel, directement lié à un composant physique, est composé d'opérations et de modèles associés à des « vues » (commande, partie opérative, contraintes et topologie). La véritable finalité du composant est son utilisation en tant que brique élémentaire assemblable pour la conception d'un système transitiqes.

Ici, nous proposons d'appliquer une telle démarche au domaine de la domotique. Un composant, dont la structure est identique à tout niveau d'abstraction, est un élément qui modélise une partie plus ou moins complexe du système domotique. À partir de composants élémentaires (*Lampes*, *Volets*, *Portes*), le concepteur peut définir des composants plus complexes de manière récursive jusqu'à l'obtention d'un système global. La stratégie d'agrégation, à la charge du concepteur, doit prendre en compte la possibilité de réutilisation de ces composants agrégés. En effet, une fois paramétrés, ces derniers peuvent être réintégrés dans la bibliothèque en perspective d'une réutilisation future. (Les chambres d'un centre de soins présentent souvent le même environnement domotique. Dès lors, après en avoir conçu une à partir de composants élémentaires, celle-ci peut-être stockée et réutilisée, en conservant certains paramètres et en en modifiant d'autres.)

4. PIM : Platform Independent Model

5. PSM : Platform Specific Model

6. KNX Association (Official website) : www.knx.org

Le modèle que nous proposons fait la distinction entre les points de vue *ressource* et *système* pour la modélisation physique d'un environnement domotique. Ils s'inspirent naturellement des concepts issus des SAP et reprend donc leur terminologie. Une *ressource* est une représentation logicielle d'un équipement domotique qui possède une ou plusieurs opérations (la ressource *Télévision* possède les opérations *Allumer*, *Eteindre*, *Couper le son*...). Une opération est en fait une fonction réalisée par une ressource spécifique [3]. Dès lors, une fonction, définie selon le point de vue *système*, peut être implantée par diverses ressources (la fonction *Allumer la lumière* peut être implantée sur la ressource *Lumière1* pour aboutir à l'opération *Allumer la Lumière1*). Les ressources sont donc logiquement mises en oeuvre par des composants reposant sur une structure standard.

À la différence des SAP, un système domotique d'assistance est avant tout dédié à la personne. Il est alors indispensable d'introduire un point de vue *utilisateur* pour séparer le modèle physique à base de composants du modèle logique prenant en compte les besoins spécifiques des utilisateurs.

4. Prise en compte des besoins de l'utilisateur

Dans cette section, nous considérons la manière dont l'utilisateur doit pouvoir interagir avec son environnement, afin de fournir au concepteur les moyens de modéliser les interactions entre l'utilisateur et l'habitat indépendamment de l'interface homme-machine qui sera utilisée. Le modèle proposé permet à l'utilisateur d'interagir avec l'environnement domotique de deux manières. La première permet d'exprimer l'injonction « *Je veux* » à travers les services. De manière complémentaire, la seconde permet d'exprimer la formule « *J'interdis* » mise en oeuvre par les modes de fonctionnement. Tout d'abord, nous considérons l'expression des interdictions, puis nous détaillons la demande de services.

4.1. Les modes de fonctionnement

Les modes de fonctionnement sont souvent cités dans les travaux liés ou apparentés aux SAP, dont l'objectif était initialement de remplacer l'action de l'homme pour les tâches dures et répétitives. D'après [4], les modes sont définis par un ensemble d'états caractérisant une ressource ou un ensemble de ressources selon un point de vue. Les états qui sont regroupés sous un même mode sont mutuellement exclusifs (par exemple, dans le mode *Fonctionnement* regroupant les états {*Normal*, *Dégradé*, *Hors service*}, un seul état peut être actif). La gestion des modes, éprouvée dans le domaine des SAP, doit cependant être modifiée pour s'appliquer à l'assistance aux personnes dépendantes. Ainsi, les modes ne sont plus définis par rapport à un objectif de production mais sont rendus flexibles pour s'adapter aux besoins de l'utilisateur, afin qu'ils permettent aux concepteurs de modéliser des interdictions au niveau d'un environnement domotique.

Dans ces travaux, nous appelons *mode* une restriction sur l'espace d'états d'une ressource. Chaque mode (point de vue *utilisateur*) est interprété par un espace d'évolution E_r d'une ressource r (point de vue *ressource*). À travers les modes, le concepteur a la possibilité de restreindre l'espace d'états global $E_r = \{e_1, e_2, e_3, e_4\}$ d'une ressource en interdisant un état, par exemple e_3 . Dans ce cas, les états $e_i \in E_r$ potentiellement accessibles d'une ressource r sont définis de la manière suivante :

$$\overline{E_r} = e_3 \Rightarrow E_r = E_r \setminus \overline{E_r} = \{e_1, e_2, e_3, e_4\} \setminus \{e_3\} = \{e_1, e_2, e_4\} \quad (1)$$

où $\overline{E_r}$ représente l'ensemble des états de E_r que l'on interdit (ou que l'on rend inaccessible) par changement du mode de fonctionnement. Par ailleurs, plusieurs restrictions sur l'espace des états peuvent être activées simultanément, par l'intermédiaire d'autant de modes. Cela permet de contraindre de manière encore plus précise l'espace d'évolution d'une ressource. Par exemple, si E_r^1 représente l'espace des états autorisés par un premier mode (par exemple *économie d'énergie*), et que E_r^2 représente l'espace des états autorisés par un second mode (par exemple *enfant sans surveillance*), alors, l'espace des états autorisés par ces deux modes se calcule par intersection de E_r^1 et de E_r^2 , comme cela est illustré sur l'exemple suivant :

$$E_r = E_r^1 \cap E_r^2 = \{e_1, e_2, e_3\} \cap \{e_2, e_3, e_4\} = \{e_2, e_3\} \quad (2)$$

Mais la définition de modes ne se limite pas seulement à des contraintes sur les ressources. Un gestionnaire permet de garantir une certaine cohérence de modes, en s'assurant non seulement

que l'état d'une ressource n'est plus atteignable mais également que la ressource en question quitte cet état, si par hasard il s'agissait de l'état courant au moment du changement de modes. Par exemple, en mode *évacuation*, le concepteur définit que les portes doivent être ouvertes. En d'autres termes, il interdit l'état *fermé* des portes : `ModeInterdit(porte, fermé)`. En conséquence, l'opération associée à la fermeture des portes n'est plus accessible. Si dans notre exemple, une porte est fermée lors du changement de modes, le système doit commander son ouverture.

4.2. Les services

Au-delà de l'expression d'interdiction sur le fonctionnement des ressources, il est nécessaire, afin de prendre en compte la diversité des besoins que les personnes à mobilité réduite peuvent exprimer [15], de proposer aussi un modèle permettant l'expression de requêtes sur ces mêmes ressources. C'est ce que nous appelons le *modèle de services*.

Pour cela, nous définissons tout d'abord un service comme une tâche que propose le système à l'utilisateur dans le but de satisfaire ses besoins. Les services sont donc définis par rapport au point de vue de l'*utilisateur*, contrairement aux opérations (point de vue *ressource*) et aux fonctions (point de vue *système*). Il semble par la suite intéressant de distinguer la manière dont le service est activé de la sémantique qui lui est associée. Cette distinction permet d'aboutir à une définition multi-dimensionnelle du modèle de services où la nature de ces derniers est définie par le couple (*condition*, *niveau sémantique*). Un service *S* peut être activé de différentes façons {*c1*, *c2*, *c3*} : sous la demande de l'utilisateur à l'exécution (*c1*), systématiquement sous une condition prédéfinie avec (*c3*) ou sans demande de confirmation de l'utilisateur (*c2*). Un changement de modes peut impliquer, à travers l'activation de type (*c2*), le déclenchement d'un service pour mettre les équipements en conformité avec ce mode. Les services *S* peuvent aussi être différenciés en fonction du niveau sémantique qu'il véhicule {*n1*, *n2*, *n3*} :

- *n1* : service élémentaire associé à l'opération d'une ressource : `UtilisateurVeut(ouvrir, portel)`
- *n2* : scénario composé de plusieurs services de niveau *n1*
- *n3* : service de haut niveau sémantique associé à une fonction : `UtilisateurVeut(silence)`

5. Le flot de conception

Dans cette partie, nous reprenons l'ensemble des parties précédentes en l'intégrant dans un flot de conception. La méthodologie générale retenue pour l'aide à la conception d'un système domotique d'assistance est donc représentée par la figure 1. Chaque étape de ce flot est commentée et illustrée pour valider de manière informelle l'approche de conception retenue dans nos travaux.

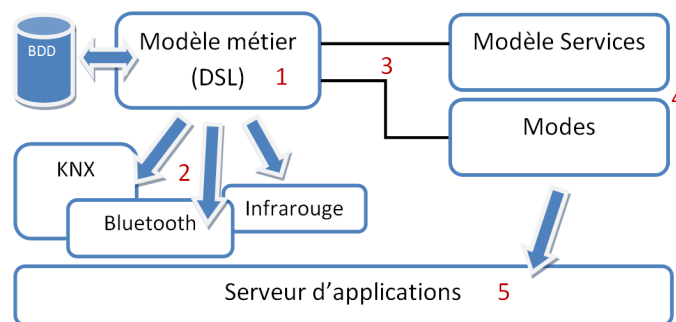


FIGURE 1 – Flot de conception pour les systèmes domotiques d'assistance à la personne

Etape 1 : Modélisation de l'environnement domotique par agrégation de composants

Le concepteur construit son environnement domotique en manipulant un modèle métier indépendant de toute plateforme. L'originalité de l'approche réside dans l'adoption d'une modélisation ascendante par agrégations successives de composants. À titre d'illustration, la figure 2 représente le plan d'un appartement que le concepteur cherche à modéliser. À partir des composants

élémentaires issus d'une bibliothèque $\{C1, TV1, \dots, L1\}$, le concepteur décide de définir un composant de plus haut niveau à l'échelle d'une pièce (*Salon*), lui-même agrégé avec d'autres composants pour former le composant *Appartement*. Ces composants agrégés, une fois définis et paramétrés, peuvent être stockés dans une bibliothèque pour une réutilisation ultérieure.

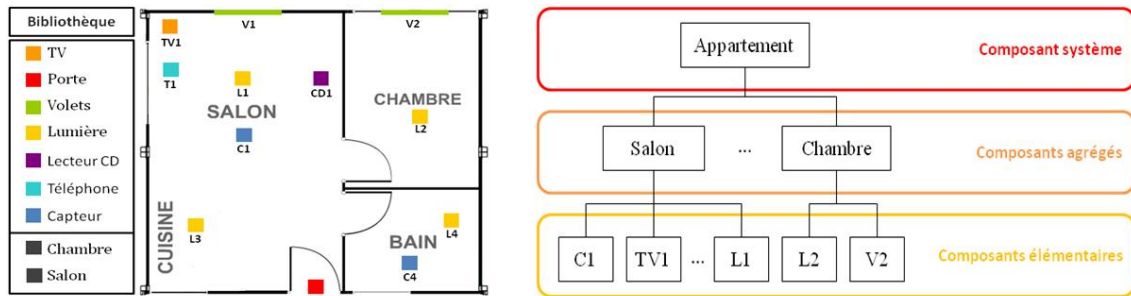


FIGURE 2 – Approche composant pour la modélisation de l'environnement domotique

Etape 2 : Génération du code de commande des équipements domotiques

Chaque composant utilisé dans l'étape 1, qu'il soit élémentaire ou agrégé, possède une structure standard et est constitué d'opérations qui lui sont propres. Par exemple le composant *L1* étant une instantiation du type *Lumière*, celui-ci possède les opérations *Allumer* et *Eteindre*. Dès lors, il est possible de générer le code de commande des différents équipements physiques à partir de la définition de chacune de leurs opérations et de leurs spécifications matérielles. Ainsi, à partir du modèle domotique correctement défini, le code de commande spécifique à une technologie (KNX-EIB, Infrarouge, Wifi...) peut être généré, de manière conforme à ce que préconise l'IDM, à savoir par une transformation d'un modèle PIM vers un modèle PSM.

Etape 3 : D'une approche composant vers une approche orientée service

Comme expliqué dans la section 4.2, les services proposés à l'utilisateur, quels que soient leurs niveaux sémantiques et leurs processus d'activation, sont décrits à l'aide d'opérations. De même, les modes de restrictions sont décrits en termes d'états atteignables ou pas. Grâce à cette description indiquant les correspondances entre opérations-services et états-modes, il est possible, dans cette étape, de passer d'un modèle métier décrivant l'architecture du système vers une représentation des services propres à l'utilisateur. Par exemple, les opérations *Allumer* et *Eteindre* de la lumière *L1* sont transformées en services élémentaires *UtilisateurVeut(allumer, lumière_L1)* et *UtilisateurVeut(éteindre, lumière_L1)*.

Etape 4 : Prise en compte des besoins de l'utilisateur

À partir des services élémentaires issus de l'étape précédente, il est possible de créer des services composites (*scenario*) et de les soumettre à d'éventuelles conditions d'activation. Il en est de même pour la définition des modes de fonctionnement. Par contre, la résolution d'un service appelé à un haut niveau sémantique (demande de l'augmentation de la lumière, ou de la diminution du bruit) est un problème plus complexe nécessitant l'utilisation de programmes d'optimisation sous contraintes, voire de méthodes issues de l'intelligence artificielle. Ces aspects sortent du cadre de cette présentation, et nous donnons quelques pistes en conclusion.

Etape 5 : Implantation des modèles pour l'exécution sur plateforme réelle

Le modèle obtenu après avoir mis en relation services et codes de commande est implanté physiquement dans un serveur permettant ainsi de publier les services domotiques et les modes préalablement définis. L'architecture matérielle retenue dans nos travaux est celle développée dans [14]. Il s'agit d'une architecture décentralisée où les serveurs communiquent entre eux pour délivrer des services. Les utilisateurs ont alors la possibilité de se connecter aux serveurs via un terminal de type PDA⁷, bien souvent embarqué dans leur fauteuil roulant, pour solliciter des services.

7. PDA : Personal Digital Assistant

6. Conclusion et Perspectives

Un flot de conception original pour les systèmes domotiques d'assistance a été présenté dans cet article. Il adopte une approche mixte de modélisation visant à satisfaire à la fois les besoins du concepteur et ceux de l'utilisateur final. La modélisation de haut niveau à l'aide d'un langage métier (DSL) ainsi que les processus de génération automatique de code rendent tout d'abord accessible la conception de tels systèmes à des personnes non-expertes. Cette modélisation architecturale à base de composants est couplée avec une approche orientée service pour répondre aux besoins des personnes à mobilité réduite. Ainsi, les modes de fonctionnement, couplés avec des services de différents niveaux sémantiques, permettent d'offrir un modèle complet de services domotiques pour améliorer l'autonomie de ces personnes. L'architecture MDA sur laquelle repose ce flot de conception offre de plus une solution flexible et évolutive pour l'intégration de nouvelles technologies domotiques. Le point que nous n'avons pas abordé concerne la résolution de services de haut niveau sémantique. Pour cela, nous pensons que la modélisation des interactions par l'utilisation d'ontologies est intéressante. Ces dernières permettent la description et le partage de l'information en explicitant les liens sémantiques entre les ressources, ce qui permet la manipulation des concepts en fonction de ces structures sémantiques, comme cela apparaît dans les nouveaux services web [7]. Enfin, à plus long terme, un déploiement à échelle réelle est envisagé au CMRRF de Kerpape⁸ afin d'évaluer les travaux présentés dans cet article.

Bibliographie

1. Engineering Tool Software. <http://www.knx.org/be-fr/knx-tools/ets/intro>.
2. Lonmaker. <http://www.echelon.com/Products/networktools/lonmaker>.
3. P. Berruet. *Contribution au recouvrement des systèmes flexibles de production manufacturière : analyse de la tolérance et reconfiguration*. PhD thesis, UST de Lille, 1998.
4. N. Dangoumau. *Contribution à la gestion des modes des systèmes automatisés de production*. PhD thesis, Université des Sciences et Technologies de Lille, 2000.
5. C. Szyperski et al. Component software : beyond object-oriented programming, 2002.
6. F. Paas et al. Cognitive load theory : Instructional implications of the interaction between information structures and cognitive architecture. *Instructional Science*, 32(1) :1–8, 2004.
7. H. Chen et al. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03) :197–207, 2004.
8. M. Jimenez et al. Habitation : a domain-specific language for home automation. *IEEE Software*, 26(4) :30–38, 2009.
9. N. Kushwaha et al. An intelligent agent for ubiquitous computing environments : smart home UT-AGENT. 2004.
10. N. Noury et al. Nouvelles directions en habitats intelligents pour la santé : New trends in health smart homes. *ITBM-RBM*, 24(3) :122–135, 2003.
11. T. Herfet et al. EMBASSI multimodal assistance for infotainment and service infrastructures. *Computers & Graphics*, 25(4) :581–592, 2001.
12. T.B.T. Truong et al. Proactive remote healthcare based on multimedia and home automation services. In *Proceedings of the fifth IEEE CASE*, pages 385–390, 2009.
13. J.-L. Lallican. *Proposition d'une approche composant pour la conception de la commande des systèmes transitiques*. PhD thesis, Université de Bretagne-Sud, 2007.
14. S. Lankri. *Services et Navigation pour Personnes Dépendantes en Environnement Domotique*. PhD thesis, Université de Bretagne-Sud, 2009.
15. *Projet QUATRA : Rapport QuatRA Final*, 2006.
16. O. M. S. Classification internationale du fonctionnement, du handicap et de la santé. 2001.
17. M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7) :75–84, 1993.

Les auteurs remercient la région Bretagne qui finance ces travaux de thèse, ainsi qu'un des rapporteurs anonymes dont les questions pertinentes ont permis d'enrichir notre réflexion à plus long terme sur les développements de ce projet.

8. CMRRF : Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelle de Kerpape