Conception d'un système domotique pour l'assistance aux personnes dépendantes

Willy Allègre[*] [**] — Thomas Burger[*] — Pascal Berruet[*] — Jean-Paul Departe[**]

[*] *Laboratoire Lab-STICC (CNRS) - Université de Bretagne-Sud - Centre de recherche - BP 92116 - F-56321 Lorient Cedex*

{prenom.nom}@univ-ubs.fr

[**] *Laboratoire d'électronique - CMRRF de Kerpape - BP 78 56275 Ploemeur Cedex*

jp.departe@kerpape.mutualite56.fr

RÉSUMÉ. *Les personnes en situation de handicap éprouvent parfois de réelles difficultés, voire une incapacité physique totale, à effectuer les activités de la vie quotidienne de manière autonome. De nombreuses initiatives ont introduit la domotique et les habitats intelligents comme une solution possible pour compenser ce handicap. 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 dépendantes. 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 contrôle-commande domotique.*

ABSTRACT. *People with disabilities sometimes have considerable difficulties, or even physical incapacities, performing daily tasks independently. Many research works have introduced home automation as a useful way to overcome these activity limitations. 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 the disabled. 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.*

MOTS-CLÉS : *domotique, ingénierie dirigée par les modèles, systèmes pervasifs*

KEYWORDS: *home automation, model driven engineering, ambient assisted living.*

1. Introduction

Les personnes en situation de handicap ont parfois des difficultés, voire même une incapacité totale à effectuer les activités quotidiennes de manière autonome. En effet, que cette situation soit liée à l'âge ou au handicap lui-même, ces personnes présentent des capacités motrices réduites et/ou des troubles cognitifs ne permettant pas de contrôler efficacement la multitude d'équipements présents dans leur environnement.

Le problème de la dépendance étant fortement lié à l'âge, le vieillissement continuel de la population dans les pays développés tend vers une augmentation du nombre de personnes dépendantes. (En 2050, 1/3 de la population sera âgé de plus de 60 ans en Europe (Lutz *et al.*, 2008).) Le nombre d'aidants potentiels ne pouvant pas évoluer en conséquence, la prise en charge de ces personnes dans le cadre du maintien à domicile constitue un enjeu sociétal et économique majeur dans les années à venir.

La nouvelle définition du handicap (Schneidert *et al.*, 2003) proposée par l'*Organisation Mondiale de la Santé* inclut les facteurs contextuels pour prendre en compte l'impact de l'environnement sur un individu. Ainsi, la situation de handicap n'est plus seulement définie par les capacités intrinsèques de la personne, mais également par l'environnement dans lequel il évolue. Il est donc possible d'améliorer la situation de handicap d'une personne en adaptant l'environnement à ses besoins, et augmenter ainsi son degré d'autonomie.

Longtemps cantonné à des aspects médicaux, la prise en charge des personnes dépendantes peut à présent bénéficier des avancées des technologies de l'information et de la communication. En ce sens, la domotique est une solution possible pour permettre de prolonger le temps où les personnes vivent dans leur environnement préféré, retardant ainsi l'échéance du placement en institution. 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 : on parle d'*habitat intelligent*.

La domotique et les habitats intelligents rejoignent le domaine de l'informatique ubiquitaire et des systèmes pervasifs (Weiser, 1993) composés d'objets communicants capables de traiter de l'information et de proposer des services de manière indépendante. (Les termes d'*informatique pénétrante* ou d'*intelligence ambiante* sont aussi utilisés.) Du fait du nombre important d'équipements potentiellement "intelligents" présents dans un environnement domotique, il devient complexe pour les utilisateurs d'en assurer le contrôle et le suivi. 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 diverses informations contextuelles, pour adapter dynamiquement le processus de composition de services aux besoins de l'utilisateur (Kushwaha *et al.*, 2004). D'autres travaux ont vu le jour pour la composition de services de haut niveau : on parle "d'applications dirigées par les buts" (Herfet *et al.*, 2001). Les besoins exprimés par l'utilisateur sont traduits par un objectif de haut niveau que le système décompose en actions élémentaires pour résoudre ce dernier.

Ces initiatives témoignent d'une certaine dynamique autour de la thématique des habitats intelligents pour l'assistance aux personnes dépendantes (Noury *et al.*, 2003; Gaucher, 2007). Néanmoins, très peu de ces réalisations se sont intéressées à la conception de tels systèmes (configuration des équipements et définition de services) et à leurs caractéristiques originales :

– **Adaptabilité** : les services domotiques et multimédia doivent répondre aux besoins particuliers des utilisateurs, intimement liés à leur situation de handicap, mais aussi à la configuration de leur espace de vie.

– **Flexibilité** : alors que la majorité des solutions domotiques sont restreintes à une utilisation individuelle, il est nécessaire d'adresser des environnements divers et multi-utilisateurs à grande échelle (e.g. centre de soins).

– **Disponibilité** : en lien étroit avec la sécurité, la sûreté de fonctionnement et la reconfiguration en cas de panne constituent deux caractéristiques essentielles qui doivent être prises en compte pour assurer la délivrance d'un service.

– **Interopérabilité** : il faut faire face à l'hétérogénéité des technologies et protocoles de communication présents dans un habitat, ainsi que celles qui seront à intégrer.

– **Évolutivité** : de nouveaux protocoles de "haut niveau" doivent pouvoir être intégrés sans avoir à modifier le travail de conception initial.

À cela, s'ajoute un certain nombre de critères qui interviennent dans l'acceptation d'une solution domotique :

– **Prix** : il comprend le coût de l'installation électrique, des logiciels de supervision et de l'intervention d'un expert pour la configuration et l'adaptation du système.

– **Intrusivité** : si les caméras et autres capteurs permettent de recueillir des informations pertinentes pour le *monitoring* d'activités ou pour adapter un comportement "intelligent", leur présence ne doivent en aucun cas perturber l'intimité d'un individu.

– **Facilité de déploiement** : la configuration doit être accessible à des personnes non-expertes en domotique pour adapter le système à l'utilisateur dépendant dont les besoins peuvent évoluer rapidement.

Actuellement, un domoticien est chargé d'installer, d'intégrer, puis de configurer la solution domotique pour convenir aux besoins de l'utilisateur. À chaque fois que ce dernier a besoin de modifier cette configuration pour intégrer un nouvel équipement ou modifier un service, il fait appel à cet expert. Généralement, une tierce personne (e.g. un ergothérapeute) est chargée de recueillir les besoins de l'utilisateur et de les transmettre au domoticien.

Dans ce contexte, nous pensons que la répartition des rôles de chaque acteur du schéma de collaboration utilisateur/ergothérapeute/domoticien doit évoluer (cf figure 1). Pour des raisons financières, nous voulons tout d'abord limiter l'intervention de l'expert. Dans le cadre de l'AAL (*Ambient Assisted Living*) où les besoins de l'utilisateur évoluent en permanence, le coût relatif à l'intervention systématique d'un expert ne peut pas être supporté par tous. Pour une meilleure intégration des besoins de l'utilisateur, nous souhaitons de plus privilégier le lien de proximité entre l'utilisateur et la

tierce personne. En effet, il semble naturel que l'adaptation d'un lieu de vie pour une personne en situation de handicap soit réalisée par un ergothérapeute ou tout autre personne intimement liée à l'utilisateur (dans la confidence de ses besoins, e.g. membre de la famille), plutôt que par un domoticien.

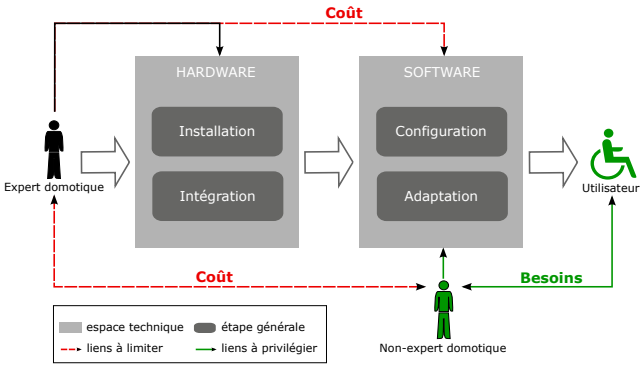

Figure 1. *Schéma de collaboration pour une solution domotique AAL*

Les quelques contributions apportées pour l'aide à la conception de tels systèmes sont bien souvent limitées du fait i) de la modélisation adoptée restreignant l'utilisation à des personnes expertes et ii) du manque d'expressivité ne permettant pas de répondre aux besoins divers et variés des personnes en situation de handicap. Pour adresser ces deux points, nous proposons :

– Une méthodologie de conception basée sur l'ingénierie dirigée par les modèles (IDM) pour faciliter l'appréhension et automatiser le processus de conception

– Un langage de description abstrait de toute spécificité matérielle et suffisamment expressif pour répondre aux besoins des utilisateurs dépendants

Après avoir positionné nos travaux par rapport aux approches de conception existantes (section 2), nous présentons la méthodologie générale de conception (section 3) pour répondre aux besoins du concepteur non-expert. Afin de tenir aussi compte du point de vue de l'utilisateur en situation de handicap, un langage de description composé de deux vues est défini à la section 4. Un flot de conception pratique déroulé sur un cas d'étude (section 5) permet enfin d'illustrer et de valider notre proposition.

2. État de l'art

Différents travaux se sont intéressés à l'aide à la conception de systèmes domotiques. Il y a tout d'abord les outils basés sur une modélisation spécifique et restreinte à une plateforme particulière : *Platform Specific Modeling* (section 2.1). Ce sont des outils de CAO (Conception Assisté par Ordinateur), bien souvent commerciaux et propriétaires, les plus répandus et utilisés en domotique. Il existe aussi des outils

basés sur une modélisation indépendante de toute plateforme (*Platform Independent Modeling*) qui permettent de s'abstraire de la cible matérielle pour ne s'intéresser qu'aux spécifications fonctionnelles, en utilisant par exemple un langage de modélisation généraliste : *General Purpose Langage* (section 2.2). Afin d'élever d'avantage le niveau d'abstraction et permettre de mieux répondre aux besoins spécifiques d'un domaine particulier, un langage de description dédié (*Domain Specific Langage*) peut s'avérer indispensable (section 2.3).

2.1. *Platform Specific Modeling*

La plupart des solutions d'aide à la conception présentes sur le marché sont spécifiques à une plateforme technologique et a fortiori à un équipementier (pour les systèmes propriétaires) ou aux associations qui supportent cette technologie (pour les systèmes standardisés).

Engineering Tool Software (KNX Association, 1993) et *LonMaker* (Echelon Corporation, 1998) sont deux des outils commerciaux les plus connus, respectivement spécifiques aux plateformes *KNX/EIB* (European Installation Bus) et *LonWorks*. Ils permettent de définir l'adressage, le paramétrage ainsi que la définition de groupes pour leurs équipements respectifs.

Malheureusement, la plupart d'entre eux sont spécifiques à une plateforme technologique, ce qui pose différentes sortes de problèmes :

– Bien que ces outils soient largement utilisés par les domoticiens, ils se restreignent à l'utilisation d'une seule plateforme technologique. Le concepteur doit alors utiliser autant d'outils qu'il y a de technologies à supporter.

– Ils utilisent de plus des procédures de bas-niveau pour le développement et la configuration d'un système domotique, ce qui rend le travail de conception exclusivement réservé à des experts en domotique.

– Enfin, les solutions logicielles sont bien souvent propriétaires et coûteuses, ce qui ne joue pas en faveur de l'acceptation d'une solution domotique.

2.2. *General Platform Independent Modeling*

Les propositions basées sur l'IDM, permet au concepteur de capturer des spécifications indépendamment de la plateforme cible (*Platform Independent Modeling*) en s'abstrayant des spécificités technologiques, garantissant ainsi l'interopérabilité d'un environnement domotique complet.

PervML (Muñoz *et al.*, 2005; Muñoz *et al.*, 2006) est une spécification UML dont le but est de fournir aux développeurs de systèmes pervasifs un ensemble de constructions leur permettant de décrire ce genre de systèmes, de manière indépendante de toute technologie. Ce langage de modélisation définit un ensemble de modèles basés

sur les vues UML (diagrammes de séquence, d'états-transitions, etc.), pour permettre aux "analystes système" et "architectes système" de décrire un système pervasif.

useML (Meixner *et al.*, 2011) se concentre sur la génération d'interface utilisateur. Un processus de conception s'assure que les besoins de ce dernier sont bien pris en compte à chaque étape de ce processus. Il est possible dans la dernière étape de ce processus, de spécifier les caractéristiques matérielles et logiciels de l'interface à concevoir, les étapes précédentes étant indépendantes de toute plateforme.

EntiMid (Nain *et al.*, 2008) propose un intergiciel, construit au dessus d'une plateforme OSGI (*Open Services Gateway Initiative*), permettant de garantir l'interoperabilité entre systèmes domotiques hétérogènes utilisant différents protocoles de communication. L'objectif de ce middleware est de garantir l'interconnexion des réseaux, l'ouverture à des protocoles de haut niveau, favoriser les accès nomades, offrir une solution dynamique de déploiement et garantir finalement la sûreté de fonctionnement.

UML est composé de centaines d'éléments génériques. Ce type de langage généraliste (*General Purpose Language*) s'éloigne des exigences du domaine AAL. Pour le concepteur non-expert en informatique, les modèles issus de ce type de langage sont complexes à manipuler, nécessitant un bagage technique avancé.

2.3. *Specific Platform Independent Modeling*

Les langages spécifiques à un domaine particulier (*Domain Specific Language*) permettent de prendre en compte exclusivement les besoins précis d'un domaine.

HomeDSL (Jimenez *et al.*, 2009; Sanchez *et al.*, 2011) est un langage graphique composé d'abstractions spécifiques au domaine de la domotique. Il permet ainsi aux automaticiens de retrouver les concepts techniques de ce domaine sous une forme générique (interopérabilité nécessaire pour le développement d'un système complet), pour par la suite générer automatiquement le code d'une ou plusieurs plateformes domotiques.

Même si de nombreux efforts ont été faits pour faciliter l'appréhension de la conception, le DSL adopte une logique automatique reprenant les concepts existants dans le domaine de la domotique : association entre capteurs, contrôleurs et actionneurs. Dans le cadre de l'assistance aux personnes dépendantes, deux problèmes peuvent être identifiés :

– le langage de description proposé manque d'expressivité lorsque l'on veut considérer les besoins divers et variés d'utilisateurs en situation de handicap (handicap physique, troubles cognitifs).

– le concepteur doit être une personne sensibilisée, voire experte en domotique pour utiliser ce DSL.

L'originalité des travaux présentés dans cet article se base sur la prise en compte de ces deux points. Afin de formaliser les étapes essentielles à suivre pour le développe-

ment d'un système domotique d'assistance, une méthode de conception est d'abord introduite.

3. Méthode de conception

3.1. *Un processus de conception centrée utilisateur*

Deux acteurs peuvent être identifiés dans nos travaux. Il s'agit du **concepteur** et de l'**utilisateur**. Dans le cadre de l'AAL, ces deux acteurs présentent des profils originaux par rapport au contexte standart d'utilisation de la domotique.

L'utilisateur peut présenter des capacités motrices réduites ou des troubles cognitifs (voire une combinaison des deux), à des stades plus ou moins avancés. Des interactions de haut niveau sémantique sont alors essentielles pour deux raisons :

– Le contrôle efficace de plusieurs équipements simultanément devient parfois difficile à assurer pour l'utilisateur. En proposant des interactions intuitives, la charge cognitive [1] peut être réduite : "Sortir" présente un niveau sémantique supérieur à "Ouvrir la porte", et de ce fait facilite la compréhension et la prise de décision pour certains utilisateurs.

– Mais ces interactions permettent surtout de compenser le handicap physique en facilitant (adaptation en termes d'interface de contrôle) ou en automatisant le fonctionnement de l'habitat : "Sortir" permet l'activation de plusieurs commandes comme "Ouvrir la porte", "Éteindre les lumières" regroupés sous une unique action.

Le concepteur, quant à lui, n'est plus un technicien mais une personne non experte en domotique, chargée de définir un environnement de vie adapté à une personne en situation de handicap. C'est donc avant tout une personne capable de prendre en compte les besoins exprimés par ce dernier. Dans le cadre institutionnel, le concepteur peut faire partie du personnel soignant (e.g. un ergothérapeute), alors que dans le cadre du maintien à domicile, il peut simplement s'agir d'un membre de la famille. Un système domotique d'assistance doit être facile à concevoir, facile à maintenir, facile à commander et flexible. La personne non experte a donc besoin d'outils de conception/modélisation adaptés pour :

– ne s'intéresser qu'aux spécifications fonctionnelles plutôt qu'aux problèmes d'implémentation afin de se focaliser sur les besoins spécifiques de l'utilisateur.

– générer automatiquement le code des différentes plateformes pour faire face à la diversité des technologies domotiques investiguées, et garantir la sureté de fonctionnement (éviter les erreurs liées aux opérateurs).

1. La charge cognitive peut devenir excessive à cause de la gestion simultanée de plusieurs équipements et du manque d'automatisation (intrinsic load) ou de la façon dont les informations sont représentées (extraneous load) (Paas *et al.*, 2004).

Les besoins originaux présentés ci-dessus peuvent être résumés ainsi : il faut élever le niveau d'abstraction, que ce soit au niveau de la conception ou de l'utilisation du système domotique. Pour cela, nous proposons une méthodologie intuitive (cf figure 2) basée sur un processus de conception centrée utilisateur (Maguire, 2001) (*Human Centered Design*, ISO 13407). Une fois les besoins de l'utilisateur recueillis, le concepteur (idéalement ergothérapeute ou membre de la famille) spécifie le contexte d'utilisation. À partir de ces spécifications, il peut adapter cet environnement pour répondre aux besoins de l'utilisateur sans être obligé de systématiquement faire intervenir un technicien. Ces deux dernières étapes permettent de définir des solutions adaptées qui seront par la suite évaluées conjointement par le concepteur et l'utilisateur. Dans le cas où le système ne convient pas ou plus (évolution des besoins) à l'utilisateur, le processus de conception est réitéré.

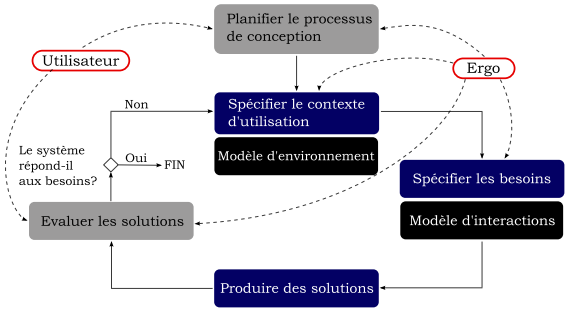

Figure 2. *Processus de conception centré utilisateur*

Afin de permettre la mise en oeuvre informatique d'un tel processus, nous utilisons l'IDM pour mettre à disposition du concepteur des modèles indépendants de toute plateforme, lui permettant ainsi d'assurer les étapes de spécification du contexte d'utilisation (modèle d'environnement) et de spécification des besoins de l'utilisateur (modèle d'interactions). Les processus de transformation de modèles permettent de plus de garantir la transition entre les différentes étapes de la méthodologie et de produire les solutions / implémentations logicielles conformément aux spécifications préalablement définies.

3.2. *Une méthodologie de conception dirigée par les modèles*

L'Ingénierie Dirigée par les Modèles (IDM) se distingue des différentes approches issues du génie logiciel. Sa démarche pose comme principe que tout ou partie d'une application est généré à partir de modèles. Le code source d'une application n'est donc plus considéré comme l'élément central d'un logiciel, mais comme un élément dérivé d'éléments de modélisation. Cela permet alors de réduire la complexité des systèmes informatiques tant pour la conception que la validation du logiciel.

Cette approche est basée sur trois concepts fondamentaux : le modèle, le méta-modèle et la transformation de modèles. Un modèle est une représentation partielle d'un système physique défini à partir d'un langage de modélisation, qu'il soit spécifique à un domaine particulier comme un DSL ou bien qu'il soit conforme à une norme reconnue (e.g. UML). Le méta-modèle définit la structure du langage (i.e. sa grammaire) utilisée pour construire ce dernier : tout modèle est conforme à son méta-modèle. Enfin, la transformation de modèles définit des règles pour traduire un modèle source en un ou plusieurs modèles cibles.

Nous pensons que l'utilisation conjointe d'un DSL et d'une architecture dirigée par les modèles permet de rendre plus accessible la conception de systèmes domotiques d'assistance. L'utilisation d'un langage dédié permet tout d'abord de répondre à des besoins particuliers d'un domaine, en ne conservant que les concepts métier qui font sens dans ce domaine. Bien que l'utilisation d'UML en tant que langage de modélisation soit largement répandue, et convienne pour exprimer les spécifications de nombreux systèmes, il ne semble pas approprié dans le cadre du domaine de l'AAL. L'utilisation de *profiles* permet de combler ces brèches en définissant des concepts spécifiques à un domaine. Mais il reste relativement difficile de supprimer les concepts génériques du langage spécialisé, qui doit de plus se restreindre à la sémantique UML. Nous utilisons également l'architecture MDA (*Model Driven Architecture*) (Miller *et al.*, 2001), spécification de l'OMG (*Object Management Group*), pour séparer explicitement les spécifications fonctionnelles (décrites par un *Platform Independent Model* (PIM)) de l'architecture d'exécution (représentée par un *Platform Specific Model* (PSM)). En élevant ainsi le niveau d'abstraction, le concepteur non-expert en domotique se focalise sur les spécifications fonctionnelles sans se préoccuper de l'implémentation du système. L'implémentation est par la suite automatisée à travers l'utilisation de transformations de modèles. La génération automatique de code de commande pour différentes technologies est un point essentiel dans le processus de conception d'un système domotique : il faut prendre en compte la diversité des protocoles et équipements présents (KNX/EIB, infrarouge, bluetooth, etc). Les transformations de modèles y apportent une solution en automatisant le passage d'un modèle indépendant défini à l'aide d'un DSL vers un modèle spécifique à une plateforme particulière, pour finalement générer le code implémentable. Enfin, l'architecture MDA offre une solution flexible et évolutive quant à l'intégration de nouvelles technologies domotiques.

La méthodologie de conception retenue adopte donc une architecture dirigée par les modèles (cf figure 3). À partir de PIM décrivant les spécifications liées i) à l'environnement et ii) aux interactions utilisateur-système, un ou plusieurs PSM sont automatiquement générés à l'aide de transformations de modèles, pour finalement générer le code de contrôle-commande implémentable sur une plateforme particulière. Le concepteur, qui a la possibilité de manipuler les modèles indépendants de toute plateforme, se décharge alors des étapes de génération de code de contrôle commande spécifique à une plateforme dont l'utilisateur peut par la suite bénéficier.

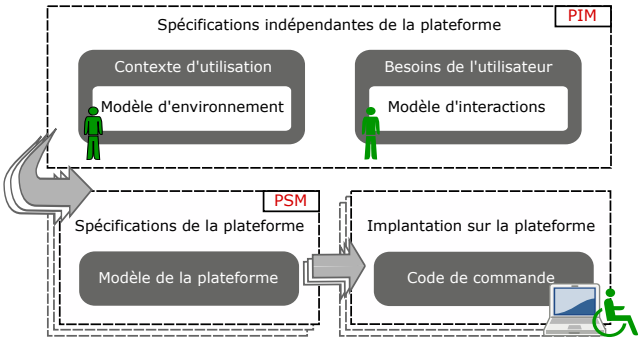

Figure 3. *Flot de conception général dirigé par les modèles*

La définition des modèles indépendants de toute plateforme est rendue possible grâce à l'utilisation d'un langage dédié aux systèmes domotiques d'assistance. Il doit être suffisamment intuitif et expressif pour permettre 1) à un concepteur non-expert en domotique d'appréhender la conception et 2) de répondre aux besoins de l'utilisateur en situation de handicap.

4. Langage de description

Le langage de description proposé comprend deux vues : l'une permet de définir les spécifications du point de vue système (modélisation de l'environnement), l'autre répond aux besoins de l'utilisateur en permettant de modéliser les interactions utilisateur-système.

4.1. *Modélisation de l'environnement basée sur les composants*

Par environnement domotique, nous entendons un habitat dans lequel l'utilisateur évolue. Qu'il soit défini à l'échelle d'un appartement individuel ou bien d'un centre de soins, il est constitué de pièces, d'équipements standards (e.g. volets, portes) et multimédia (e.g. TV, PC).

L'approche composant introduite dans ces travaux fait référence à la définition d'un composant issue du paradigme CBSE (*Component Based Software Engineering*) qui définit un composant comme une "boite noire" qui fournit uniquement ses interfaces et ses exigences (Szyperski *et al.*, 2002). Un composant est considéré comme :

– **substituable** : il peut être remplacé par un autre composant à partir du moment où ce dernier possède la même interface. *Il est possible par exemple de changer une lampe infrarouge par une lampe contrôlable par un bus domotique, à partir du moment où elle présente les mêmes fonctionnalités.*

– **composable** : il est possible de les assembler pour former des composants de plus haut niveau et aboutir ainsi à des fonctionnalités plus complexes. *À partir des composants élémentaires "capteur" et "lumière", un composant peut être défini pour obtenir une commande de plus haut niveau : "allumer_auto" qui lorsque le capteur détecte une présence, active la lumière.*

– **réutilisable** : il peut être stocké dans une bibliothèque en vue d'être réutilisé pour une autre application. *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 certians paramètres et en modifiant d'autres.*

Dans nos travaux, à chaque composant logiciel est associé un composant physique. En d'autres termes, il n'existe pas de composants spécifiés à partir du DSL qui n'ait pas de représentation physique réelle. Cette unité de conception, qui semble relativement intuitive pour une personne non experte, est à la base du processus de conception d'un environnement domotique.

En effet, la véritable finalité du composant est son utilisation en tant que brique élémentaire assemblable pour la conception d'un système domotique formé d'un ensemble de composants. Le design pattern "composite" est au coeur de la modélisation de l'environnement domotique (cf figure 4) : un composant peut être élémentaire (lumières, portes, volets, etc) ou bien agrégé. Ce dernier peut être composé d'un ou plusieurs composants, qu'ils soient élémentaires ou agrégés. Quelque soit son niveau d'abstraction, un composant représente une partie plus ou moins complexe du système et est composé d'un nom (**name**), et de noeuds d'activation (**activation_node**) renseignant sur l'accessibilité topologique du composant.

– Un composant élémentaire doit de plus être renseigné sur le protocole de communication utilisé : **typeOfProtocol** (*Bluetooth*, *KNX/EIB* ou *Infrared*). Il s'agit de la seule information d'implémentation à renseigner pour le concepteur à ce niveau.

– Les composants agrégés sont associés entre eux par des liens topologiques (**topological_link**). Les liens topologiques ainsi que l'accessibilité des composants sont importants car leur positionnement géographique relatif a une influence sur la délivrance de services.

La méthodologie de conception prône une approche ascendante (bottom-up) de conception. À partir de composants élémentaires (représentant des équipements domotiques) issus d'une bibliothèque, le concepteur peut définir des composants plus complexes de manière récursive jusqu'à l'obtention du système global. Cette approche ascendante de conception se justifie par le fait que la définition d'un composant de niveau N se fait en ayant connaissance des opérations du composant de niveau $N-1$, pour envisager d'y associer des services de plus haut niveau. La stratégie d'agrégation, à la charge du concepteur, doit de plus prendre en compte la possibilité de réutilisation de ces composants agrégés. En effet, une fois paramétrés, ces derniers peuvent être intégrés dans la bibliothèque en perspective d'une réutilisation future.

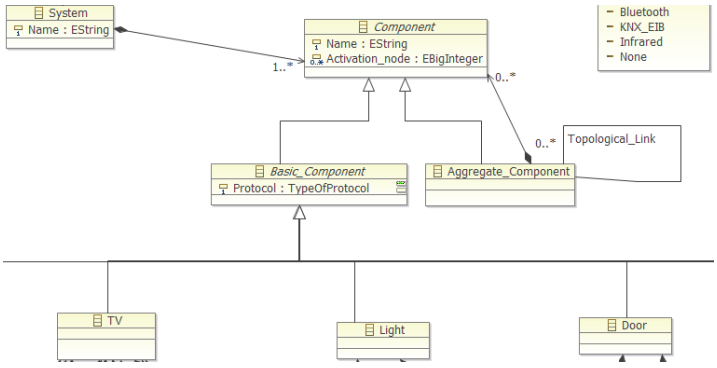

Figure 4. *Extrait du méta-modèle Environnement*

Il peut sembler curieux de proposer une approche ascendante à un non-expert, une approche descendante semblant plus adaptée à l'absence de compétence technique en domotique. Cependant, dans notre cas, cela ne pose pas de problèmes, puisque la méthodologie de conception affranchit le concepteur de ces aspects techniques/domotiques. Dès lors, l'approche ascendante ne considère que la topologie physique du cadre de vie, ce qui reste à la portée du concepteur.

Le modèle d'environnement que nous proposons fait la distinction entre les deux points de vue *ressource* et *système* (cf tableau 1). Selon le point de vue **ressource**, une **opération** permet de changer les états d'une ressource particulière. Cette dernière est une représentation logicielle d'un équipement domotique qui comporte une ou plusieurs opérations. Par exemple, la ressource *TV1* possède plusieurs opérations, comme *Allumer TV1*, *Éteindre TV1*, etc. Pour les raisons expliquées ci-dessous, les états de chaque ressource peuvent aussi être soumis à des **restrictions**. Les ressources sont logiquement implémentées par des composants basés sur une structure standart.

Ressource	**Système**	**Utilisateur**
Opération	Fonction	Service : "Je veux"
Restriction	Héritage	Mode : "J'interdis"

Tableau 1. *Points de vue Ressource/Système/Utilisateur et concepts associés*

Le point de vue **système**, défini pour élever le niveau d'abstraction, est responsable de la supervision du système domotique. Selon ce point de vue, une **fonction** est défini par un concept abstrait, qui lorqu'implémentée par une ressource, fournit une opération. De plus, une fonction, peut être implémentée par plusieurs ressources. Par exemple, le fonction *Allumer la lumière* peut être implémentée sur la ressource *Lumière1* pour former l'opération *Allumer la lumière 1*. Les restrictions définies au niveau ressource sont par la suite transmises par héritage au niveau système.

À la différence des SAP, les systèmes domotiques d'assitance sont avant tout dédiés à la personne. C'est pouquoi nous introduisons un troisième point de vue, celui de l'**utilisateur**, dans le but de séparer la modélisation physique basée sur les composants du modèle logique qui considère les besoins spécifiques de l'utilisateur. Les opérations et les fonctions sont exprimées en terme de **service**, alors que les restrictions sont exprimées en terme de **mode**.

La structure hiérarchique à laquelle aboutit le concepteur permet d'associer des interactions à différents niveaux du système. Cela laisse par exemple la possibilité de définir le contrôle d'une lumière particulière, de l'ensemble des lumières d'une pièce, ou encore du système complet. Ces interactions sont modélisées à travers le point de vue utilisateur, détaillé dans la section suivante.

4.2. *Modélisation des interactions utilisateur-système*

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.

4.2.1. *Les modes*

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 (point de vue *ressource*). À travers les modes, le concepteur a la possibilité de restreindre l'espace d'états global $E_{Global}(r) = \{e_1, ..., e_N\}$ d'une ressource r, en interdisant un ensemble d'états :

$$\begin{array}{l} E_{Interdit}(r) = \{e'_1, ..., e'_M\},\ M < N \\ E_{Interdit}(r) \subset E_{Global}(r) \end{array} \quad [1]$$

Dans ce cas, les états potentiellement accessibles de r sont définis par :

$$E_{Access}(r) = E_{Global}(r) \setminus E_{Interdit}(r) \quad [2]$$

Par ailleurs, plusieurs restrictions sur l'espace d'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 $E^1_{Access}r$ représente l'espace des états autorisés par un premier mode (par exemple *économie d'energie*), et que $E^2_{Access}r$ 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 $E^1_{Access}r$ et de $E^2_{Access}r$.

$$E_{Access}(r) = E_{Global}(r) \setminus \bigcup_{i=0}^{k} E^i_{Interdit}(r) = \bigcap_{i=0}^{k} E^i_{Access}(r) \quad [3]$$

La gestion des modes, approuvée dans le domaine des SAP, a été adaptée pour être appliquée au domaine de la domotique pour l'assistance aux personnes en situation de handicap. D'après (Hamani *et al.*, 2006), 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). Ici, 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.

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 : *SystInterdit(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.

De même, l'activation d'un mode n'est autorisé que si la ressource a au moins un état potentiellement activable. Si l'on considère une porte *porte1* ayant les états *ouvert*, *fermé* et les modes *"AntiFugue" = SystInterdit(porte1, ouvert)* et *"Évacuation" = SystInterdit(porte1, fermé)*, le gestionnaire de modes interdit la co-existence de ces deux modes au même moment.

4.2.2. *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 (De Lamotte *et al.*, 2008), 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 $\{c1, c2, c3\}$:

– $c1$: sous la demande de l'utilisateur

– $c2$: automatiquement sous une condition prédéfinie

– $c3$: automatiquement avec confirmation de l'utilisateur

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. Si dans l'exemple précédent, une porte est fermée lors de l'activation du mode *évacuation*, le système va demander à l'utilisateur s'il veut ouvrir cette porte. Les modes peuvent de plus être soumis aux même type de condition d'activation que les services.

Les services S peuvent aussi être différenciés en fonction du niveau sémantique qu'il véhicule $\{n1, n2, n3\}$:

– $n1$: service élémentaire associé à une opération

– $n2$: scénario composé de plusieurs services $n1$

– $n3$: service de haut niveau sémantique associé à une fonction

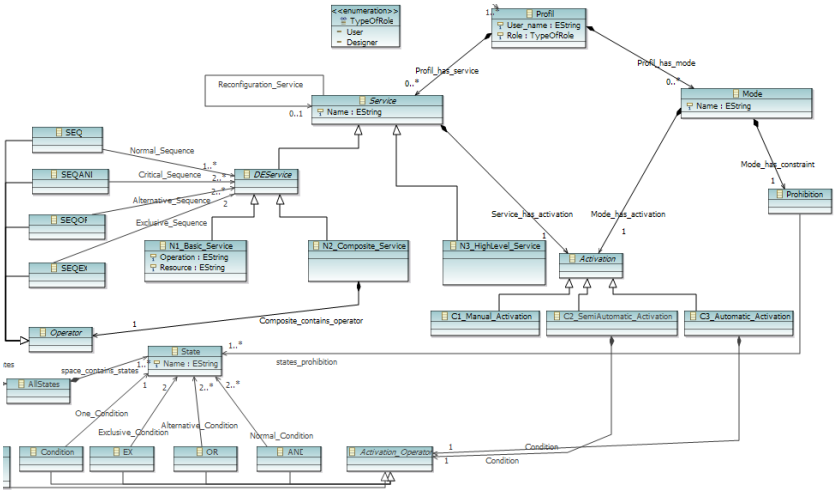

Figure 5. *Extrait du méta-modèle Interaction*

La figure 5 reprend l'ensemble des interactions pouvant être décrites à partir du DSL. Les scénarios sont définis à l'aide d'opérateurs permettant de composer différents services à évènement discret (**DEService**). La figure 6 illustre le com-

portement des opérateurs à l'exécution en utilisant le formalisme des Réseaux de Petri (Petri, 1962). L'opérateur de séquence **SEQ** permet d'exécuter tous les services (N1_Basic_Service ou N2_Composite_Service) dans l'ordre et considère que la séquence s'est déroulée correctement quoiqu'il arrive. **SEQAND** permet à une séquence de s'arrêter lorsqu'une exécution d'un service élémentaire renvoie une erreur. **SEQOR** est un opérateur de recouvrement prenant en compte deux services exclusivement. Il permet, dans le cas d'une erreur d'exécution du premier service, d'exécuter le deuxième. Enfin, l'opérateur **SEQEX** a la même sémantique que l'opérateur précédent mais est considéré comme critique dans le sens où l'échec du premier service engendre une exécution erroné de l'ensemble du scénario.

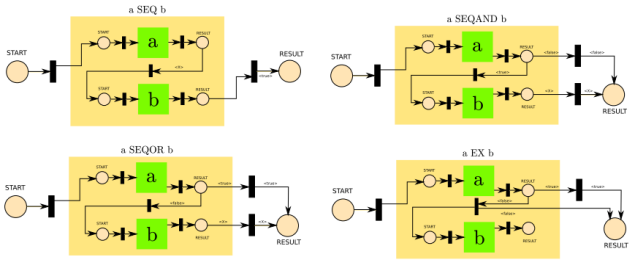

Figure 6. *Opérateurs de composition pour les scénarios*

Les opérateurs de composition, en plus de l'automatisation, permettent de prendre en compte le retour d'exécution des scénarios. Lorsque ce dernier échoue, des solutions de reconfiguration (**Reconfiguration_Service**) peuvent être mise en place. Par exemple, le concepteur peut envisager l'activation d'une lumière alternative dans le cas où la lumière principale d'une pièce est défectueuse.

Tout ou partie de ces définitions sont illustrées dans la section suivante.

5. Mise en oeuvre sur un cas d'étude

5.1. *Description du cas d'étude*

L'utilisateur de ce cas d'étude est une personne devenue hémiplégique il y a quelques mois après un AVC [2].

Ce dernier présente des capacités motrices réduites (difficultés pour la marche et membre supérieur droit non fonctionnel) ne lui permettant pas de réaliser les tâches de la vie courante sans l'assistance systématique d'une tierce personne (soignant, aide

2. Accident Vasculaire Cérébral : première cause de handicap physique chez l'adulte aux États-unis et en Europe. Le handicap (physique ou cognitif), affecte 75% des victimes d'AVC (Diener *et al.*, 2009).

à domicile, membre de la famille, etc). Se déplaçant en fauteuil roulant, sa mobilité réduite ne lui permet de marcher que pour réaliser les transferts lit/fauteuil et vice versa. La domotique représente alors pour cette personne une solution pour faciliter, voire automatiser le fonctionnement de son habitat.

Le sujet présente également des troubles cognitifs débutants dans le cadre d'une démence de type Alzheimer [3]. Pour l'instant, il s'agit surtout de troubles mnésiques légers (petits oublis...). Progressivement, ces troubles de mémoire vont s'accentuer, entraînant une désorientation temporo-spatiale, et plus tard des fugues. L'environnement domotique, en réponse à ces troubles, doit alors être capable de s'adapter pour répondre correctement aux besoins émergeants.

Le patient a d'abord été hospitalisé dans une Unité Neuro-Vasculaire avant d'effectuer plusieurs mois de rééducation dans un centre de Soins de Suite et Réadaptation (SSR) en hospitalisation complète puis de jour. Il rentre maintenant au domicile avec des aides techniques et humaines. Son habitat est à présent équipé d'un bus domotique et de nombreux équipements pouvant être commandés par infrarouge. Dans la suite de ce cas d'étude, nous considérons uniquement la modélisation de l'habitat personnel de l'utilisateur.

5.2. *Description de la plateforme cible*

DANAH (Lankri *et al.*, 2008) est la plateforme domotique cible retenue dans nos travaux. Il s'agit d'un middleware dédié aux systèmes domotiques d'assistance qui permet le contrôle environnemental mais aussi l'aide à la navigation de fauteuils. Destiné à être déployé sur des bornes domotiques distribuées dans l'environnement, ce middleware tire profit des travaux de la communauté Linux (Allègre *et al.*, 2011b) en utilisant des librairies libres et open-sources pour la prise en charge de différents protocoles de communication (infrarouge via LIRC, KNX/EIB via eibd (Kastner *et al.*, n.d.), bluetooth via la librairie BlueZ).

L'architecture de déploiement développée (figure 7) est basée sur le paradigme client-serveur. Les bornes domotiques, matérialisées par des cartes à faible consommation (ISEE, 2010), sont distribuées dans l'environnement. Une borne est ainsi connectée à un bus domotique (i.e KNX-EIB), alors que les autres sont chargées de contrôler les équipements à activation locale (i.e. infrarouge, bluetooth). L'utilisateur, équipé d'un PDA (Personal Digital Assistant) embarqué dans son fauteuil, a la possibilité de solliciter les services et les modes publiés sur son IHM (Interface Homme-Machine).

Ouvert et flexible, ce middleware a pu être développé pour intégrer les concepts présentés dans cet article (i.e. modèle de services, modes, etc.).

3. La maladie d'Alzheimer est la forme la plus commune de démence. Elle affecte environ 26 millions de personnes dans le monde (Brookmeyer *et al.*, 2007) diagnostiquée avant l'AVC. Des études récentes montrent un lien entre cette démence et les AVC (Nagai *et al.*, 2009).

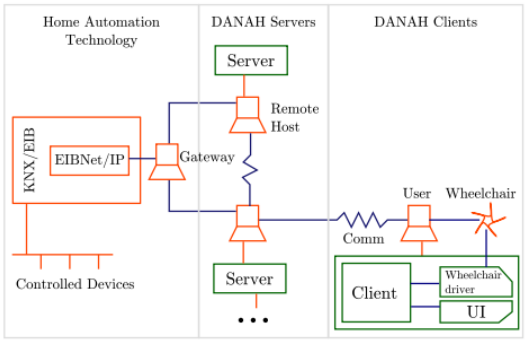

Figure 7. *Architecture de déploiement du middleware Danah*

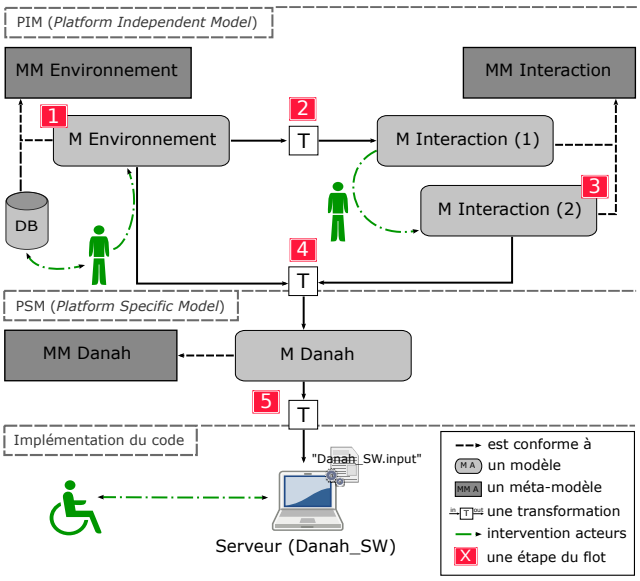

Figure 8. *Flot de conception pratique intégrant le PSM Danah*

5.3. *Flot de conception pratique*

Le flot de conception dirigé par les modèles (Allègre *et al.*, 2011a) adopte logiquement cette plateforme (cf figure 8). À partir de la définition d'un modèle d'environnement (étape 1), un modèle minimaliste d'interactions est généré à l'aide d'une transformation de modèles (étape 2) afin de spécifier les interactions permettant de répon-

dre aux besoins de l'utilisateur (étape 3). Ces deux modèles, décrits à l'aide du DSL indépendant de toute plateforme, sont ensuite transformés en un modèle spécifique à une plateforme technologique (ici, DANAH) (étape 4) pour finalement générer le code implémentable sur cette plateforme logicielle (étape 5).

Un plug-in Eclipse a été réalisé pour outiller ce flot de conception. L'édition et la manipulation se fait à l'aide de EMF (*Eclipse Modeling Framework*) et GMF (*Graphical Modeling Framework*) pour la représentation graphique du DSL. Le langage de méta-modélisation utilisé est ECore/EMOF (OMG, 2006), pour la définition du DSL (*MMEnvironment* et *MMInteraction*). Les transformations de modèles sont implémentées quant à elles en utilisant le langage ATL (*Atlas Transformation Language*) (Jouault *et al.*, 2006).

Chaque étape de ce flot est commentée et illustrée en prenant le cas d'étude de la section 5.1 pour vérifier l'applicabilité de la méthodologie de conception retenue dans nos travaux.

5.4. *Déroulement des étapes du flot*

5.4.1. *Étape 1 : Modélisation de l'environnement*

L'utilisateur est sur le point de rentrer chez lui pour la première fois depuis son AVC. Son fils, personne référente lors du retour à domicile, est chargé de modéliser l'habitat récemment équipé d'un bus domotique KNX/EIB (cf figure 9).

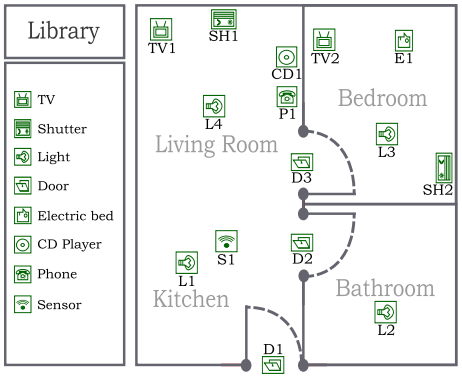

Figure 9. *Plan de l'appartement du cas détude*

L'approche de modélisation préconisée étant une approche ascendante de modélisation (des composants élémentaires jusqu'au système complet), le premier travail du concepteur est de définir les propriétés et les opérations des composants élémentaires (*D1*, *L1*, *S1*, etc.) issus de la bibliothèque.

Le concepteur choisit ensuite d'assembler ces composants pour former des composants de plus haut niveau, à l'échelle d'une pièce : les composants *Living room*, *BathRoom*, *Kitchen* sont ainsi créés. Pour intégrer les portes dans la structure hiérarchique de l'appartement, le concepteur définit trois composants agrégés (*T1*, *T2* et *T3*) dits "de transition" qu'il va lier (avec l'association **topological_link**) avec les composants précédemment définis (ces derniers doivent également renseigner un composant de transition avec l'association **topological_node**). Ainsi, le composant agrégé *T2* sera composé de la porte *P2* et associé aux composants agrégés *Bathroom* et *Kitchen*. Enfin, ces composants sont regroupés pour former le composant *Apartment*. Il en résulte une structure hiérarchique composé de **23** composants et **71** opérations associées représentant l'environnement domotique, tous liés par des relations de composition ou d'association (cf figure 10).

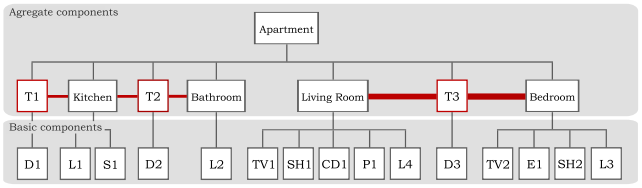

Figure 10. *Représentation hiérarchique du modèle d'environnment (PIM)*

5.4.2. *Étape 2 : 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 restriction 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 décrivant l'architecture du système vers une représentation de services propres à l'utilisateur. Ce changement de vue se traduit par l'utilisation de la transformation de modèles *Env2Int* dont la principale règle de transformation concerne la transformation d'opérations en services élémentaires (*c1*, *n1*), décrite par l'algorithme de la figure 11, et implémenté en utilisant le langage ATL (cf figure 12).

FOR ALL Basic_Components
  FOR ALL Operations O
    DO Create service S **WHERE**
      $S.name = O.name$
      $S.condition = C1$
      $S.semanticlevel = N1$
  END FOR
END FOR

Figure 11. *Représentation partielle algorithmique de Env2Int*

```
rule operation2service
  from
    s : MMEnvironment !Operation
  to
    t : MMInteraction !C1_Manual_Activation,
    u : MMInteraction !N1_Basic_Service (
        Name ← s.name,
        Service_has_activation ← t
    )
```

Figure 12. *Représentation partielle ATL de Env2Int*

Les opérations *Allumer* et *Eteindre* de la lumière *L1* sont par exemple transformées en services élémentaires soumis à une condition d'activation manuelle. Il en est de même pour les **71** opérations du modèle d'environnement. Les états associés à chaque opération sont également transformés pour aboutir à un modèle d'interaction minimaliste comprenant une liste de 71 états qu'il sera par la suite possible d'interdire et autant de services élémentaires que le concepteur pourra enrichir.

5.4.3. *Étape 3 : Enrichissement du modèle d'interactions*

À partir des services élémentaires, il est possible de créer des scénarios et de les soumettre à des conditions d'activation. De la même manière, les modes peuvent être soumis à des conditions particulières.

Dans un premier temps, l'utilisateur a besoin d'automatisation pour compenser son handicap physique. Lorsqu'il était en rééducation, l'ergothérapeute, accompagné de l'équipe médicale et paramédicale, a pu définir des scénarios $S(c1, n2)$ usuels à l'aide d'opérateurs de composition pour son retour à domicile (cf figure 6) :

– "Je dors" permet de descendre le lit électrique *E1*, d'éteindre la *TV2*, et d'éteindre la lumière *L3* : *UtilisateurVeut(E1.Down and TV2.Power and L3.SwitchOFF)*

– "Urgence" utilise un téléphone infrarouge qui permet d'appeler un premier numéro (*Key3*). Dans le cas où cet appel se solde par un échec, un second numéro est appellé (*Key4*), puis la porte *D1* s'ouvre : *UtilisateurVeut((P1.Key3 or P1.Key4) and D1.Open)*

Afin d'automatiser d'avantage le fonctionnement de l'habitat, le concepteur soumet des services (élémentaires ou scénarios) à des conditions d'activation automatique ($S(c3, ni)$) :

– "L1 auto" permet d'allumer la lumière *L1* lorsqu'une présence est détectée (capteur S1) : *(S1.ON)* $\Rightarrow$ *SystExécute(L1.SwitchON)*

– "Silence" permet de mettre sur *Mute* les équipements *TV1*, *TV2* et le lecteur *CD1* lorsque le téléphone *P1* est en communication ou bien est décroché : *(P1.PickUP or P1.Dial)* $\Rightarrow$ *SystExécute(TV1.Mute or TV2.Mute or CD1.Pause)*

Mais les besoins de l'utilisateur évoluent quelques mois après son retour à domicile. En effet, la démence est maintenant au stade modéré avec une augmentation des troubles mnésiques entraînant une légère désorientation temporo-spatiale. Pour pallier ces nouveaux troubles, la personne référente, en lien avec l'ergothérapeute, décide de définir des services qui nécessitent une demande de confirmation de la part de l'utilisateur. Ces services, rythmant la journée, l'aident à se repérer dans le temps et stimulent sa mémoire :

– "Dormir" permet de demander à l'utilisateur, lorqu'il est 23h, s'il veut activer le scénario "Je dors" préalablement défini : *(Time = 23h)* $\Rightarrow$ *SystDemande("Je dors")*

– "Jour" permet de demander à l'utilisateur, lorqu'il est 9h, s'il veut monter les volets *SH1* et *SH2* : *(Time = 9h)* $\Rightarrow$ *SystDemande(SH1.up and SH2.up)*

– "SerieTV" permet de demander à l'utilisateur, lorqu'il est 14h, s'il veut regarder sa serie préféré (chaîne 1 de la *TV1*) : *(Time = 14h)* $\Rightarrow$ *SystDemande(TV1.Ch1)*

– "Nuit" permet de demander à l'utilisateur s'il veut descendre ces volets *SH1* et *SH2* lorsqu'il est 21h : *(Time = 21h)* $\Rightarrow$ *SystDemande(SH1.down and SH2.down)*

La démence de type Alzheimer est à présent au stade sévère. De manière complémentaire aux services d'automatisation et en accord avec l'équipe médicale, le fils de l'utilisateur en situation de handicap définit un certain nombre d'interdiction sur l'environnement domotique dans le but de prévenir d'une éventuelle fugue et assurer la sécurité de ce dernier. Un mode Antifugue est alors défini, interdisant l'ouverture de la porte d'entrée pendant la nuit :

– "Antifugue" permet d'interdire l'ouverture de la porte *D1* à partir de 22h, et ceux jusqu'à 7h le lendemain : *(Time* $\geq$ *22h and Time* $\leq$ *7h)* $\Rightarrow$ *SystInterdit(D1.Open)*

À ce stade, on va aussi privilégier l'automatisation des services qui étaient précédemment activables manuellement ou automatiquement avec demande de confirmation, permettant ainsi de retarder l'échéance du placement en institution.

5.4.4. *Étape 4 : Du DSL au modèle spécifique à la plateforme Danah*

À partir des modèles indépendants de toute plateforme définis précédemment, il est maintenant possible de générer un modèle spécifique à la plateforme domotique Danah conformément à son méta-modèle qui a été défini pour être intégré dans le flot de conception.

Ce dernier est constitué de deux niveaux (**unit** et **action**), que l'on peut retrouver sur l'IHM de l'application cliente. Les **unit** sont composés entre autre d'un module **device** permettant de renseigner les spécificités matérielles du composant :

– **protocol** et **uri** décrivent respectivement le protocole à charger pour ce composant et l'emplacement de la librairie associée

– le module **option** contient l'**id** et **value** permettant de renseigner l'identifiant, l'adresse du composant sur le bus ou sur un réseau, suivant le protocole utilisé

La transformation PIM vers PSM, considère les deux modèles définis à partir des

étapes 1 et 3. C'est l'ensemble des informations présentes dans ces deux modèles qui vont permettre de définir le modèle spécifique à la plateforme Danah.

– Le modèle d'environnement permet de générer les informations (**protocol**, **uri**, **id**, **value**) relatives au protocole de communication utilisé par un composant converti en **unit**. (Par exemple, *L1* étant défini comme un composant utilisant le protocole KNX/EIB, un unit comprenant le code de commande et la librairie associée est automatiquement généré.)

– Le modèle d'interactions permet quant à lui de générer des actions comprenant d'éventuelles conditions d'activation (**condition**) et de définir les paramètres **expression** ou **exec** suivant le type de service considéré, respectivement un service composite/mode ou un service élémentaire. Danah étant structuré en deux niveaux (unit et action), des units sont définis pour regrouper les scénarios et les modes.

À ce niveau, l'installateur ou une personne sensibilisée à la domotique doit effectuer un *mapping* entre les opérations de chaque composant défini dans le modèle d'environnement, et une liste de commandes présente dans un fichier de configuration. Par exemple, pour des équipements *KNX/EIB*, le fichier de configuration est issu du logiciel *ETS* (KNX Association, 1993).

5.4.5. *Étape 5 : Du modèle Danah au fichier de configuration*

À partir du PSM issu de l'étape précédente, il est possible de générer du code conforme à la syntaxe Danah. Ce processus est assuré par une transformation de modèles en texte. La figure 13 reprend la description de la lampe *L1*, donnant ainsi un extrait des **892** lignes de code Danah générées pour ce cas d'étude.

Le fichier de configuration "*Danah_SW.input*" ainsi généré est alors embarqué dans un serveur domotique. L'utilisateur peut donc s'y connecter pour solliciter les interactions précédemment définies. La figure 14 donne un aperçu de l'interface graphique présente sur le PDA de l'utilisateur.

6. Conclusion et Perspectives

Un flot de conception original pour les systèmes domotiques d'assistance a été présenté dans cet article. Conformément au processus de conception centré utilisateur, 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

```
unit "1005" {
        name = "L1"
        activation_nodes = "{1254 ;3128}"

        action "SwitchON" {
                confirmation = false
                exec = "1"
                icon = "light_on.png"
        }

        action "SwitchOFF" {
                confirmation = false
                exec = "0"
                icon = "light_off.png"
        }

        property "icon" {
                icon = "light_off.png"
        }

        effect {
                pre = "L1@Status==SwitchOFF"
                post = "L1@icon==’light_off.png’"
        }

        effect {
                pre = "L1@Status==SwitchON"
                post = "L1@icon==’light_on.png’"
        }

        device {
                protocol = "libeibd.so.0"
                uri = "ip :127.0.0.1"

                option = "EIBAddr" {
                        value = "6/0/10"
                }
        }
}
```

Figure 13. *Extrait du code Danah automatiquement généré*

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. Suite aux expérimentations menées sur l’accessibilité de l’outil de CAO, des développements vont être menés pour améliorer l’ergonomie de l’interface et la mettre finalement à disposition d’ergothérapeutes du CMMRF de Kerpape [4]. Un déploiement à échelle réelle est envisagé dans les appartements témoins du centre afin d’évaluer la solution domotique complète présentée dans cet article.

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

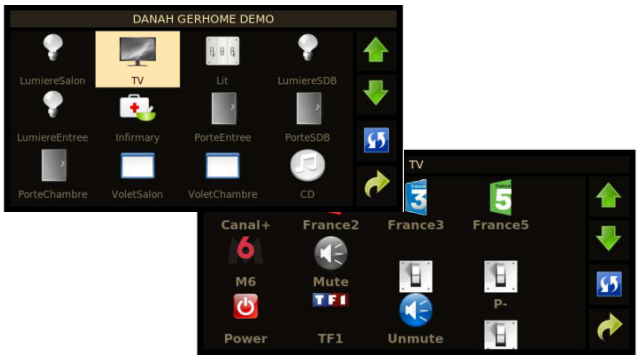

Figure 14. *Interface utilisateur Danah affichÃ©e sur le PDA*

Remerciements

Ce travail est financé par la région Bretagne et est le résultat d'une collaboration supportée par l'entreprise Vity Technology (`www.vity.com`) et le CMRRF de Kerpape (`www.kerpape.mutualite56.fr`).

7. Bibliographie

Allègre W., Berruet P., Burger T., « Model driven flow for assistive home automation system design », *IFAC World Congress 2011*, 2011a.

Allègre W., Seguin C., Burger T., De Lamotte F., Berruet P., Philippe J., Diguet J., « Ambient Assisted Living with Linux », *Proceedings of the 1st Embed With Linux (EWiLi) Workshop*, ACM, 2011b.

Brookmeyer R., Johnson E., Ziegler-Graham K., Arrighi H., « Forecasting the global burden of Alzheimer's disease », *Alzheimers Dement*, vol. 3, n° 3, p. 186-191, 2007.

De Lamotte F., Departe J., Le Saout F., Diguet J., Philippe J., « Quatra : Final report (Technical report, in french). Lab-STICC », , See demo here : http ://www.youtube.com/watch ?v=T6GCFnkLTc0, June, 2008.

Diener H., Wong P., « Developments in Secondary Stroke Prevention », , European Neurological Review - Volume 3 Issue 2, 2009.

Echelon Corporation, « LonMaker Integration Tool », , Official website : http ://www.echelon.com/, 1998.

Gaucher P., « Habitat intelligent et handicap : contexte de rÃ©flexion », *Sciences et Technologies pour le Handicap*p. 95-114, 2007.

Hamani N., Dangoumau N., Craye E., « A functional modeling approach for mode handling of flexible manufacturing systems », *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing*, p. 271-276, 2006.

Herfet T., Kirste T., Schnaider M., « EMBASSI multimodal assistance for infotainment and service infrastructures », *Computers & Graphics*, vol. 25, n° 4, p. 581-592, August, 2001.

ISEE, « IGEPv2 Board », , Official website : http ://www.igep.es/, 2010.

Jimenez M., Rosique F., Sanchez P., Alvarez B., Iborra A., « Habitation : a domain-specific language for home automation », *IEEE Software*, vol. 26, n° 4, p. 30-38, 2009.

Jouault F., Allilaire F., Bézivin J., Kurtev I., Valduriez P., « ATL : a QVT-like transformation language », *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, ACM, p. 720, 2006.

Kastner W., Neugschwandtner G., K
"ogler M., « An open approach to EIB/KNX software development », Citeseer, n.d.

KNX Association, « Engineering Tool Software », , Official website : http ://www.knx.org/, 1993.

Kushwaha N., Kim M., Kim D., Cho W., « An intelligent agent for ubiquitous computing environments : smart home UT-AGENT », 2004.

Lankri S., Berruet P., Rossi A., Philippe J., « Architecture and models of the DANAH assistive system », *Proceedings of the 3rd international workshop on Services integration in pervasive environments*, ACM, p. 19-24, 2008.

Lutz W., Sanderson W., Scherbov S., « The coming acceleration of global population ageing », *Nature*, vol. 451, n° 7179, p. 716-719, 2008.

Maguire M., « Methods to support human-centred design », *International journal of human-computer studies*, vol. 55, n° 4, p. 587-634, 2001.

Meixner G., Seissler M., Breiner K., « Model-Driven Useware Engineering », *Model-Driven Development of Advanced User Interfaces*p. 1-26, 2011.

Miller J., Mukerji J. et al., « Model driven architecture (mda) », *Object Management Group, Draft Specification ormsc/2001-07-01*, 2001.

Muñoz J., Pelechano V., « Building a software factory for pervasive systems development », *Advanced Information Systems Engineering*, Springer, p. 342-356, 2005.

Muñoz J., Serral E., Cetina C., Pelechano V., « Applying a Model-Driven Method to the Development of a Pervasive Meeting Room », *ERCIM News*, vol. 65, p. 44-45, 2006.

Nagai M., Hoshide S., Kario K., « Hypertension and dementia », *American journal of hypertension*, vol. 23, n° 2, p. 116-124, 2009.

Nain G., Daubert E., Barais O., Jézéquel J., « Using mde to build a schizophrenic middleware for home/building automation », *Towards a Service-Based Internet*p. 49-61, 2008.

Noury N., Virone G., Ye J., Rialle V., Demongeot J., « New trends in health smart homes », *ITBM-RBM (RBM)*, vol. 24, n° 3, p. 122-135, 2003.

OMG, « MOF 2.0 core specification », , http ://www.omg.org/spec/MOF/2.0, 2006.

Paas F., Renkl A., Sweller J., « Cognitive load theory : Instructional implications of the interaction between information structures and cognitive architecture », *Instructional Science*, vol. 32, n° 1, p. 1-8, 2004.

Petri C. A., Kommunikation mit Automaten, PhD thesis, Institut für instrumentelle Mathematik, Bonn, 1962.

Sanchez P., Jimenez M., Rosique F., Alvarez B., Iborra A., « A framework for developing home automation systems : from requirements to code », *Journal of Systems and Software*, 2011.

Schneidert M., Hurst R., Miller J., Üstün B., « The role of environment in the International Classification of Functioning, Disability and Health (ICF) », *Disability & Rehabilitation*, vol. 25, n° 11, p. 588-595, 2003.

Szyperski C., Gruntz D., Murer S., « Component software : beyond object-oriented programming. 2002 », 2002.

Weiser M., « Some computer science issues in ubiquitous computing », *Communications of the ACM*, vol. 36, p. 75-84, 1993.

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél. : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com