



# Flot de conception dirigé par les modèles pour la commande et la supervision de systèmes domotiques d'assistance

Willy Allègre

► **To cite this version:**

Willy Allègre. Flot de conception dirigé par les modèles pour la commande et la supervision de systèmes domotiques d'assistance. Automatique. Université de Bretagne Sud, 2012. Français. <tel-00803402>

**HAL Id: tel-00803402**

**<https://tel.archives-ouvertes.fr/tel-00803402>**

Submitted on 21 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THESE / UNIVERSITE DE BRETAGNE-SUD  
UFR Sciences et Sciences de l'Ingénieur  
Sous le sceau de l'Université Européenne de Bretagne

Pour obtenir le grade de :  
DOCTEUR DE L'UNIVERSITE DE BRETAGNE-SUD  
Mention : STIC

présenté par :

**Willy ALLEGRE**  
Lab-STICC

## **Flot de conception dirigé par les modèles pour la commande et la supervision de systèmes domotiques d'assistance**

**Thèse soutenue le 7 décembre 2012,**  
devant la commission d'examen composée de :

**M. Philippe Gorce**  
Professeur, Université du Sud Toulon / Rapporteur

**M. Eric Zamai**  
MCF-HDR, Université de Grenoble / Rapporteur

**M. Sylvain Giroux**  
Professeur, Université de Sherbrooke / Examineur

**M. André Thépaut**  
Professeur, Télécom Bretagne / Examineur

**M. Pascal Berruet**  
Professeur, Université de Bretagne Sud / Directeur de thèse

**M. Thomas Burger**  
Chercheur CNRS, CEA / Encadrant de thèse

**M. Jean-Yves Antoine**  
Professeur, Université de Tours / Invité

**M. Jean-Paul Departe**  
Ingénieur, CMRRF de Kerpage / Invité



## Résumé

En France, comme dans l'ensemble des pays développés, le même constat est dressé : la population vieillit. Cette évolution démographique sans précédent dans l'histoire de l'humanité pose un certain nombre d'enjeux économiques et sociaux liés à la prise en charge des personnes âgées dépendantes et/ou en situation de handicap, dont le nombre augmente considérablement au fil des ans. Le nombre d'aidants potentiels ne pouvant augmenter en conséquence, il faut penser dès aujourd'hui à des solutions alternatives pour permettre à ces personnes de garder un niveau d'autonomie suffisant leur permettant de rester dans leur environnement domestique préféré le plus longtemps possible. Les systèmes domotiques, qui regroupent l'ensemble des technologies de l'informatique, de l'électronique et des télécommunications au sein d'un habitat, permet d'y apporter des solutions technologiques.

Dans ces travaux, nous proposons un flot de conception dirigé par les modèles pour la commande et la supervision de systèmes domotiques d'assistance. Dans un premier temps, nous cherchons à intégrer une personne non-experte en domotique (e.g. membre de la famille / aidant, ergothérapeute) dans certaines étapes de la conception pour une meilleure prise en compte des besoins de personnes en situation de handicap mais aussi pour limiter l'intervention relativement coûteuse de l'expert. Pour cela, nous proposons un langage de modélisation (*DSML : Domain Specific Modeling Language*) permettant de décrire un système domotique en manipulant des concepts à un haut niveau d'abstraction. Ce DSML est composé de deux vues : une vue physique pour représenter l'environnement domotique et une vue fonctionnelle pour définir les interactions entre l'utilisateur et son système domotique. Dans cette dernière vue, deux types d'interactions sont définis pour prendre en compte un nombre de besoins utilisateur le plus grand possible. Ainsi, il est possible de définir des requêtes de différents types (i.e. services) comme des interdictions temporaires (i.e. modes) sur des ressources domotiques. À partir des modèles définis à l'aide de ce langage, notre flot de conception permet de générer la partie commande à l'aide de transformations de modèles successives. Une évaluation sur le cas d'étude des appartements tremplins du centre de Kerpape avec des étudiants puis des ergothérapeutes permet d'avoir des retours intéressants sur l'utilisabilité du DSML et du flot de conception proposé.

Dans un deuxième temps, nous cherchons à offrir une solution alternative aux approches de supervision basées sur l'utilisation massive de capteurs, qui peut parfois être problématique lorsque l'on s'intéresse à l'acceptabilité de la solution par l'utilisateur (coût, intrusivité). Nous proposons pour cela d'intégrer dans le flot de conception un modèle de supervision non-intrusif basé sur une ontologie pour représenter les interactions utilisateur-système. Contrairement aux solutions existantes, ce modèle n'est pas basé sur des données de capteurs, mais sur l'analyse sémantique des commandes domotiques envoyées par l'utilisateur. L'ontologie et les règles qui lui sont associées permettent à un moteur d'inférences d'extraire des informations habituellement captées : suivant la manière dont l'utilisateur interagit avec son système domotique, sa localisation et son intention peuvent être déduites par inférences logiques, permettant d'aboutir à la proposition de services contextuels. Une évaluation sur les jeux de données de l'appartement Domus permet de montrer l'intérêt d'une telle approche tant pour limiter l'utilisation de capteurs que pour augmenter la sûreté de fonctionnement par redondance d'informations. Une partie de ces travaux est reprise dans le cadre du projet ASIM porté par l'entreprise Vity Technology. La syntaxe concrète (i.e. représentation graphique) du langage de modélisation sera notamment retravaillé dans un but de commercialisation.

## Abstract

Population ageing is widespread across the world. Unprecedented in the history of mankind, this demographic trend leads to a number of social and economic issues related to ageing/disabled people whose number increases considerably over the years. As the number of caregivers can not evolve accordingly, we must now think of alternatives allowing these people to stay longer in their preferred domestic environment with a sufficient level of autonomy. Home automation, which includes computer, electronics and telecommunications within a housing, allows to provide technological solutions.

In this work, we propose a model-driven flow for assistive home automation systems. First, we seek to integrate a home automation non-expert (e.g. family member / caregiver, occupational therapist) in the design process to better take into account the needs of people with disabilities, but also to limit the costly intervention of the expert on site. To do so, we propose a modeling language (*DSML: Domain Specific Modeling Language*) allowing one to describe a home automation system by manipulating high-level abstraction concepts. This DSML is composed of two views: the environment view to model the home automation environment and the interaction view to model user-system interactions. In this latter, two types of interactions are defined to consider the greatest number of needs. Thus, it is possible to define requests (i.e. services) and temporary prohibitions (i.e. modes) on home automation devices. From models defined with this language, our design flow allows to generate the control part through successive model transformations. An evaluation on the Kerpape Smart Home case study with students and occupational therapists provides interesting results on both the proposed design flow and the DSML usability.

Then, we seek to offer an alternative to context-aware approaches based on the massive deployment of sensors, which can sometimes be problematic when one focuses on the user acceptability (cost, intrusivity). We propose to integrate in the design flow a context-aware model based on ontology to define user-system interactions. Unlike existing solutions, this model is not based on sensor data, but on the semantic analysis of each user request on home automation devices. The ontology and associated rules allow an inference engine to extract information usually sensed: according to the way the user interacts with his home automation system, his location and his intention can be deduced by logic reasoning, leading to contextual services proposition. An evaluation on the Domus Smart Home dataset shows the interest of such an approach to limit the use of sensors, but also to increase the dependability by information redundancy. Part of this work is integrated in the ASIM project supported by Vity Technology. The concrete syntax (i.e. graphical representation) of the modeling language will especially be considered in the purpose of a commercialization.



n d'ordre : 000000000

**Université de Bretagne Sud**

Centre de Recherche C. Huygens - rue de Saint-Maudé - BP 92116 56231 LORIENT CEDEX

Tél : + 33(0)2 97 87 45 45 Fax : + 33(0)2 97 85 45 00



*À mes parents, le juste retour des choses.  
À Claire, pour son soutien, ses encouragements et bien plus encore...*



## Remerciements

Mes premiers remerciements vont naturellement aux personnes qui m'ont encadré durant ces trois années de thèse. Tout d'abord, je remercie Thomas Burger, chercheur au CNRS (encadrant de thèse), pour son soutien au quotidien, son enthousiasme et sa rigueur scientifique. Je remercie aussi Pascal Berruet, professeur et vice-président de l'Université de Bretagne Sud (directeur de thèse) pour la supervision de ce projet tant sur le plan humain que technique et pour la valorisation de ces travaux dans le milieu industriel. À leurs côtés, j'ai énormément appris.

J'étends mes remerciements aux membres du laboratoire Lab-STICC de Lorient, en particulier aux personnes impliquées dans l'équipe "Systèmes pervasifs pour l'aide au handicap" avec qui j'ai eu le plaisir de travailler ces trois années. Une mention spéciale à Marwa, Thanh, Linfeng, Cédric, Jean-Christophe et Ronan avec qui j'ai partagé bien plus qu'un espace de travail : une vie de "thésard" au quotidien.

Je remercie les membres du jury : Philippe Gorce, professeur à l'Université du Sud Toulon et Éric Zamai, maître de conférences à l'Université de Grenoble, qui m'ont fait l'honneur et le plaisir d'être mes rapporteurs ainsi que Sylvain Giroux, professeur à l'Université de Sherbrooke et André Thépaut, professeur à Télécom Bretagne de figurer en tant qu'examineurs.

Je tiens aussi à remercier Jean-Yves Antoine, professeur à l'Université de Tours, qui m'a apporté une aide précieuse, notamment pour l'évaluation de mes travaux auprès d'étudiants mais aussi par le financement de deux stages permettant d'en avancer le développement logiciel. Avant tout, Jean-Yves a été un interlocuteur privilégié avec qui j'ai eu le plaisir d'échanger.

Enfin, je tiens à remercier tout particulièrement Jean-Paul Departe, ingénieur au laboratoire d'électronique du CMRRF de Kerpape, pour m'avoir suivi et aidé tout au long de cette thèse. Si mon stage de Master au centre de Kerpape avait déjà été extrêmement enrichissant pour moi, je ne m'imaginai pas que cela allait guider la suite de mon parcours : des premiers pas en thèse jusqu'au laboratoire d'électronique. Aujourd'hui, je suis très honoré de poursuivre mon chemin aux côtés de Jean-Paul, qui me transmet jour après jour ses connaissances et sa vue globale sur les aides techniques électroniques et informatiques. Dans ce mémoire, j'ai tenu à reprendre ce point de vue en introduisant la domotique comme une aide technique possible pour compenser une situation de handicap. Une aide parmi tant d'autres...





# Table des matières

|   |           |
|---|-----------|
| <b>Table des matières</b>                             | <b>1</b>  |
| <b>1 Contexte &amp; Positionnement</b>                | <b>9</b>  |
| 1.1 Contexte  | 11        |
| 1.1.1 Évolution démographique                         | 11        |
| 1.1.1.1 Vieillesse des populations                    | 11        |
| 1.1.1.2 Prévalence de la dépendance                   | 14        |
| 1.1.2 Processus de Production du Handicap             | 15        |
| 1.1.2.1 Introduction au modèle conceptuel             | 15        |
| 1.1.2.2 Handicap : une situation universelle          | 17        |
| 1.1.2.3 Aménagement du lieu de vie                    | 18        |
| 1.1.3 Évolution des systèmes domotiques               | 20        |
| 1.1.3.1 Émergence des systèmes pervasifs              | 20        |
| 1.1.3.2 Évolution de la domotique                     | 21        |
| 1.2 La domotique pour l'assistance à la personne      | 23        |
| 1.2.1 Description d'un système domotique d'assistance | 24        |
| 1.2.1.1 Définition(s)                                 | 24        |
| 1.2.1.2 Caractéristiques                              | 25        |
| 1.2.2 État de l'art des projets existants             | 27        |
| 1.2.2.1 Projets d'habitats intelligents               | 27        |
| 1.2.2.2 Projets du Lab-STICC / MOCS                   | 31        |
| 1.2.3 Verrous scientifiques identifiés                | 34        |
| 1.3 Motivations & Contributions                       | 35        |
| 1.3.1 Motivations                                     | 35        |
| 1.3.1.1 Conception universelle / centrée utilisateur  | 35        |
| 1.3.1.2 Nouveau schéma de collaboration               | 36        |
| 1.3.1.3 Supervision non-intrusive d'interactions      | 37        |
| 1.3.2 Contributions scientifiques                     | 38        |
| 1.3.2.1 Langage de modélisation domotique             | 38        |
| 1.3.2.2 Flot de conception dirigé par les modèles     | 39        |
| 1.3.2.3 Modèle de supervision basé sur les ontologies | 40        |
| <b>2 DSML pour la domotique d'assistance</b>          | <b>43</b> |
| 2.1 Introduction                                      | 45        |
| 2.1.1 L'Ingénierie Dirigée par les Modèles            | 45        |
| 2.1.1.1 Organisation hiérarchique des modèles         | 45        |
| 2.1.1.2 Transformation de modèles                     | 48        |
| 2.1.2 Les langages de modélisation                    | 49        |
| 2.1.2.1 Définition                                    | 49        |
| 2.1.2.2 Les langages dédiés                           | 50        |
| 2.1.2.3 Langage et MDA                                | 51        |
| 2.2 État de l'art & positionnement                    | 53        |

|          |  |           |
|----------|--|-----------|
| 2.2.1    | Modélisation spécifique à une plateforme . . . . .       | 53        |
| 2.2.2    | Modélisation généraliste indépendante . . . . .          | 53        |
| 2.2.3    | Modélisation de domaine indépendante . . . . .           | 54        |
| 2.2.4    | Choix retenu . . . . .                                   | 55        |
| 2.3      | Description du DSML . . . . .                            | 56        |
| 2.3.1    | Méta-modèle de l'environnement . . . . .                 | 56        |
| 2.3.1.1  | Modélisation de l'environnement par composants . . . . . | 56        |
| 2.3.1.2  | Une approche ascendante de conception . . . . .          | 61        |
| 2.3.2    | Méta-modèle des interactions . . . . .                   | 64        |
| 2.3.2.1  | Les services . . . . .                                   | 64        |
| 2.3.2.2  | Les modes . . . . .                                      | 67        |
| 2.3.2.3  | Activation et définition . . . . .                       | 71        |
| 2.4      | Conclusion du chapitre / contribution . . . . .          | 72        |
| <b>3</b> | <b>Flot de conception dirigé par les modèles</b>         | <b>73</b> |
| 3.1      | Introduction au flot de conception . . . . .             | 75        |
| 3.1.1    | Processus de conception centré utilisateur . . . . .     | 75        |
| 3.1.1.1  | Caractéristiques des acteurs . . . . .                   | 75        |
| 3.1.1.2  | Intégration des acteurs dans le processus . . . . .      | 76        |
| 3.1.2    | Flot de conception théorique . . . . .                   | 77        |
| 3.1.2.1  | Architecture dirigée par les modèles . . . . .           | 77        |
| 3.1.2.2  | Flot de conception proposé . . . . .                     | 78        |
| 3.2      | Implémentation du flot de conception . . . . .           | 80        |
| 3.2.1    | Intégration du PSM Danah . . . . .                       | 80        |
| 3.2.1.1  | Présentation de Danah . . . . .                          | 80        |
| 3.2.1.2  | Méta-modèle Danah . . . . .                              | 80        |
| 3.2.2    | Flot de conception pratique . . . . .                    | 82        |
| 3.2.2.1  | Résumé des étapes de conception . . . . .                | 82        |
| 3.2.2.2  | Déroulement des étapes du flot . . . . .                 | 82        |
| 3.3      | Outillage du flot de conception . . . . .                | 90        |
| 3.3.1    | IntelHome Designer . . . . .                             | 90        |
| 3.3.1.1  | Organisation du logiciel . . . . .                       | 90        |
| 3.3.1.2  | Tableau de bord . . . . .                                | 92        |
| 3.3.2    | Éditeur graphique . . . . .                              | 93        |
| 3.3.2.1  | Modèle d'environnement . . . . .                         | 93        |
| 3.3.2.2  | Modèle d'interactions . . . . .                          | 94        |
| 3.4      | Conclusion du chapitre / contribution . . . . .          | 97        |
| <b>4</b> | <b>Modèle de supervision non-intrusif</b>                | <b>99</b> |
| 4.1      | Introduction . . . . .                                   | 101       |
| 4.1.1    | Supervision non-intrusive . . . . .                      | 101       |
| 4.1.1.1  | Définition . . . . .                                     | 101       |
| 4.1.1.2  | Contexte & Enjeux . . . . .                              | 102       |
| 4.1.2    | Introduction aux ontologies . . . . .                    | 102       |

|          |  |            |
|----------|--|------------|
| 4.1.2.1  | Les Logiques de Description . . . . .                        | 102        |
| 4.1.2.2  | Les ontologies et le langage OWL . . . . .                   | 105        |
| 4.1.3    | État de l'art & Positionnement . . . . .                     | 107        |
| 4.1.3.1  | Ontologies pour les systèmes pervasifs . . . . .             | 107        |
| 4.1.3.2  | État de l'art en domotique . . . . .                         | 107        |
| 4.1.3.3  | Choix retenu . . . . .                                       | 109        |
| 4.2      | Modélisation des interactions utilisateur-système . . . . .  | 110        |
| 4.2.1    | Conceptualisation . . . . .                                  | 110        |
| 4.2.1.1  | Contexte d'application . . . . .                             | 110        |
| 4.2.1.2  | Ontologie (T-Box) . . . . .                                  | 110        |
| 4.2.2    | Analyse sémantique . . . . .                                 | 111        |
| 4.2.2.1  | Localisation de l'utilisateur . . . . .                      | 111        |
| 4.2.2.2  | Intention de l'utilisateur . . . . .                         | 112        |
| 4.2.2.3  | Proposition de services . . . . .                            | 113        |
| 4.3      | Intégration dans le flot de conception . . . . .             | 114        |
| 4.3.1    | La spécification ODM . . . . .                               | 115        |
| 4.3.1.1  | Les ontologies et l'approche MDA . . . . .                   | 115        |
| 4.3.1.2  | Ontology Definition Metamodel . . . . .                      | 116        |
| 4.3.2    | Flot de conception pour la supervision . . . . .             | 118        |
| 4.3.3    | Architecture de supervision . . . . .                        | 119        |
| 4.4      | Conclusion du chapitre / contribution . . . . .              | 120        |
| <b>5</b> | <b>Validations expérimentales</b> . . . . .                  | <b>123</b> |
| 5.1      | Évaluation du DSML et du flot de conception . . . . .        | 125        |
| 5.1.1    | Cas d'étude : Kerpape Smart Home . . . . .                   | 125        |
| 5.1.1.1  | Un tremplin vers l'indépendance . . . . .                    | 125        |
| 5.1.1.2  | Description technique des appartements . . . . .             | 126        |
| 5.1.2    | Formalisation de l'évaluation . . . . .                      | 129        |
| 5.1.2.1  | Critères d'évaluation . . . . .                              | 129        |
| 5.1.2.2  | Méthodes d'évaluation . . . . .                              | 129        |
| 5.1.2.3  | Hypothèses formulées / expérimentations contrôlées . . . . . | 130        |
| 5.1.3    | Passage à l'échelle . . . . .                                | 131        |
| 5.1.3.1  | Utilisateur du cas d'étude . . . . .                         | 131        |
| 5.1.3.2  | Déroulement des étapes du flot . . . . .                     | 132        |
| 5.1.4    | Expérimentation contrôlée / étudiants . . . . .              | 138        |
| 5.1.4.1  | Protocole d'expérimentation . . . . .                        | 138        |
| 5.1.4.2  | Résultats de l'expérimentation . . . . .                     | 139        |
| 5.1.5    | Expérimentation contrôlée / ergothérapeutes . . . . .        | 142        |
| 5.1.5.1  | Protocole d'expérimentation . . . . .                        | 142        |
| 5.1.5.2  | Résultats de l'expérimentation . . . . .                     | 142        |
| 5.2      | Validation de la supervision . . . . .                       | 143        |
| 5.2.1    | Cas d'étude : Domus Smart Home . . . . .                     | 143        |
| 5.2.2    | Application pour le suivi de personnes . . . . .             | 145        |
| 5.2.3    | Résultats sur les jeux de données . . . . .                  | 147        |

|          |  |            |
|----------|--|------------|
| 5.2.3.1  | Limiter les coûts . . . . .                                | 148        |
| 5.2.3.2  | Augmenter la sûreté de fonctionnement . . . . .            | 148        |
| <b>6</b> | <b> Conclusion et perspectives</b>                         | <b>151</b> |
| 6.1      | Résumé des contributions . . . . .                         | 151        |
| 6.1.1    | DSML et flot de conception domotique . . . . .             | 151        |
| 6.1.2    | Modèle de supervision non-intrusif . . . . .               | 152        |
| 6.2      | Perspectives à court terme . . . . .                       | 153        |
| 6.2.1    | Expérimentations contrôlées . . . . .                      | 153        |
| 6.2.2    | Syntaxe graphique pour le DSML . . . . .                   | 154        |
| 6.2.3    | IHM dynamique et contextuelle . . . . .                    | 155        |
| 6.3      | Perspectives à long terme . . . . .                        | 155        |
| 6.3.1    | Résolution de services de haut niveau sémantique . . . . . | 155        |
| 6.3.2    | QoS pour le déploiement de capteurs . . . . .              | 156        |
| 6.3.3    | Aide technique mobile unifiée . . . . .                    | 157        |
| 6.4      | Le mot de la fin . . . . .                                 | 158        |
|          | <b>Bibliographie</b>                                       | <b>168</b> |
|          | <b>Publications personnelles</b>                           | <b>169</b> |
|          | <b>Liste des Figures</b>                                   | <b>171</b> |
|          | <b>Liste des Tableaux</b>                                  | <b>173</b> |
|          | <b>Annexes :</b>   | <b>173</b> |
| <b>A</b> | <b> Installation du logiciel</b>                           | <b>175</b> |
| <b>B</b> | <b> Exportation de fichiers</b>                            | <b>179</b> |
| <b>C</b> | <b> Questionnaire</b>                                      | <b>181</b> |
| <b>D</b> | <b> Consentement éclairé</b>                               | <b>183</b> |
| <b>E</b> | <b> Tutoriels</b>  | <b>185</b> |

# Introduction générale

Dans l'histoire de l'humanité, si le XXème siècle reste celui de la croissance démographique, le XXIème siècle sera quant à lui celui du vieillissement. Ce phénomène mondial qui fait suite au *baby boom* d'après-guerre semble inéluctable tant il est inscrit dans la pyramide des âges actuelle : les personnes qui atteindront 60 ans à l'horizon 2050 sont déjà toutes nées. Inévitablement, le vieillissement de la population implique des conséquences majeures qui auront un impact sur toutes les facettes de la société, tant sociales que économiques. L'augmentation du phénomène de la dépendance est une de ces conséquences, pouvant également impliquer des situations de handicap dans lesquelles il peut parfois être difficile, voire impossible d'effectuer les activités de la vie quotidienne sans l'aide systématique d'une tierce personne.

La nouvelle définition du handicap proposée par l'*Organisation Mondiale de la Santé* inclut à présent les facteurs environnementaux : une situation de handicap ne se définit pas exclusivement par les capacités intrinsèques d'un individu, mais aussi par l'environnement qui l'entoure. En d'autres termes, l'adaptation de l'environnement peut potentiellement améliorer une situation de handicap. Longtemps limitée à des aspects médicaux, la prise en charge de personnes dépendantes peut à présent bénéficier des avancées des technologies de l'information et de la communication. La domotique est en ce sens une solution possible pour permettre de prolonger le temps où les personnes vivent dans leur environnement préféré. Pour se faire, le système doit être capable de faciliter, voire d'automatiser l'activation de certains services. Cela implique un niveau d'interaction élevé : on parle communément d'*habitats intelligents*. Si ce terme est explicite et compréhensible par le plus grand nombre, nous préférons parler de systèmes domotiques d'assistance dans ce mémoire. Ce dernier terme fait référence à l'adaptation d'un système aux besoins précis d'un utilisateur. Il ne s'agit donc pas de proposer un système tout automatisé, mais bien de compenser pour partie une situation de handicap en automatisant ou en facilitant l'activation de services en réponse à ses besoins.

Dans ce contexte, nous proposons un flot de conception original pour répondre aux besoins des deux acteurs identifiés dans ces travaux. Il s'agit du concepteur et de l'utilisateur, qui présentent tous deux des profils originaux par rapport au contexte standard d'utilisation de la domotique. L'utilisateur peut présenter des capacités motrices réduites et/ou des troubles cognitifs à des stades plus ou moins avancés : il a besoin d'un système domotique qui puisse s'adapter à ses besoins. 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 pour une personne en situation de handicap. Il s'agit donc avant tout d'une personne capable de prendre en compte les besoins exprimés par ce dernier (e.g. membre de la famille / aidant, ergothérapeute).

Dans un premier temps, nous avons cherché à intégrer le concepteur non-expert dans certaines étapes de la conception pour une meilleure prise en compte des besoins de l'utilisateur mais aussi pour limiter l'intervention relativement coûteuse de

l'expert. Pour cela, nous avons proposé un langage de modélisation (*DSML : Domain Specific Modeling Language*) permettant de décrire un système domotique en manipulant des concepts à un haut niveau d'abstraction. Ce langage se veut le plus expressif possible pour répondre à la diversité des besoins que peuvent exprimer les personnes en situation de handicap. Nous proposons d'utiliser conjointement *l'Ingénierie Dirigée par les Modèles (IDM)*, qui pose comme principe que tout ou partie d'une application est généré à partir de modèles. L'IDM vise ainsi à réduire la complexité du système domotique via l'utilisation de représentations simplifiées (concepts présents dans le DSML), mais aussi de techniques et d'outils permettant la génération automatique de la commande et de la supervision de ce système.

Dans un deuxième temps, nous avons cherché à offrir une solution alternative aux approches de supervision<sup>1</sup> basées sur l'utilisation massive de capteurs, qui peut parfois être problématique lorsque l'on s'intéresse à l'acceptabilité de la solution par l'utilisateur (coût, intrusivité). Nous proposons pour cela d'intégrer dans le flot de conception un modèle de supervision non-intrusif basé sur une ontologie pour représenter les interactions utilisateur-système. À partir de ce modèle de représentation de connaissances largement utilisé dans le web sémantique, nous proposons d'analyser chaque commande domotique envoyée par l'utilisateur. L'ontologie et les règles qui lui sont associées permettent à un moteur d'inférences d'extraire des informations habituellement captées : suivant la manière dont l'utilisateur interagit avec son système domotique, sa localisation et son intention peuvent être déduites par inférences logiques, permettant d'aboutir à la proposition de services contextuels.

Tout au long de ces travaux, nous avons essayé d'adopter un point de vue global et pragmatique pour prendre en compte les caractéristiques des différents acteurs. Tout d'abord, nous avons tenu à combler le manque de compétences techniques du concepteur non-expert que nous voulons intégrer dans le flot de conception. Nous avons également tenu à prendre en compte l'acceptabilité de la solution par l'utilisateur pour lequel nous proposons une approche de supervision non-intrusive et un langage de modélisation permettant de répondre au mieux à ses besoins.

La figure 1 illustre l'organisation générale du mémoire. Après avoir présenté le contexte des travaux et les problématiques auxquelles faire face dans le premier chapitre, trois contributions qui tentent d'y apporter des réponses sont présentées :

- Chapitre 2 : le deuxième chapitre présente le langage de modélisation pour la domotique d'assistance en réponse au besoin de conception universelle et de la prise en compte d'un maximum de besoins utilisateur.
- Chapitre 3 : le troisième chapitre présente le flot de conception dirigé par les modèles qui précise les différentes étapes à suivre pour aboutir au code de commande de systèmes domotiques.
- Chapitre 4 : le quatrième chapitre présente la supervision non-intrusive d'in-

---

1. Par supervision, nous entendons à la fois le monitoring de l'utilisateur et la proposition de services.

teractions domotiques, basée sur les ontologies et les outils du web sémantique.

Le chapitre 5 présente enfin les validations expérimentales réalisées pour valider chacune des contributions de cette thèse. La conclusion résume toutes ces parties et permet d'ouvrir vers les perspectives qui découlent de ces travaux.

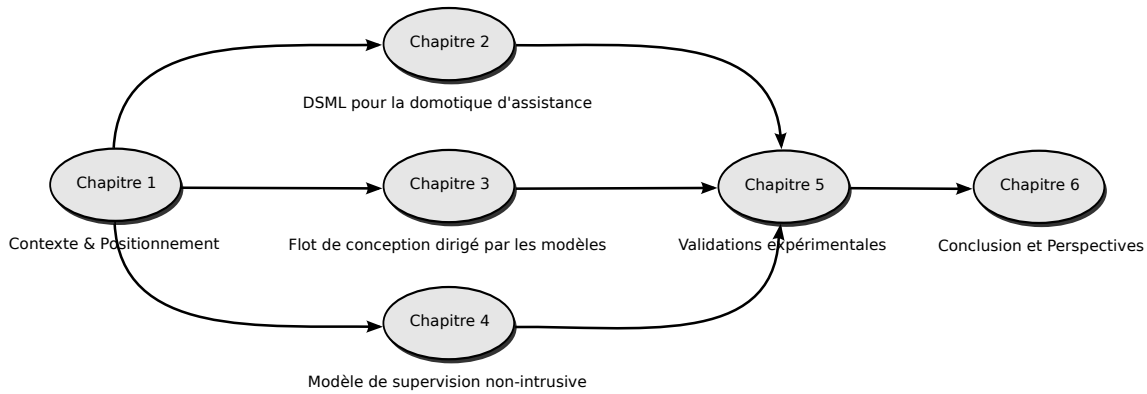


FIGURE 1 – Organisation générale du mémoire (décomposition en chapitres)





# 1

## Contexte & Positionnement

*“L’avenir n’est jamais que du présent à mettre en ordre.  
Tu n’as pas à le prévoir, mais à le permettre.”*

Antoine de Saint-Exupéry, Citadelle

Le XXI<sup>ème</sup> siècle sera incontestablement marqué par le vieillissement des populations, un phénomène inéluctable qui touche l’ensemble des pays développés. Cette évolution démographique sans précédent pose un certain nombre d’enjeux sociétaux et économiques liés à la prise en charge des personnes âgées dépendantes dont le nombre augmentera considérablement au fil des ans. Le nombre d’aidants potentiels ne pouvant augmenter en conséquence, il faut penser dès aujourd’hui à des solutions alternatives pour permettre à ces personnes de garder un niveau d’autonomie suffisant pour qu’elles puissent rester dans leur environnement domestique préféré le plus longtemps possible. La domotique, qui regroupe l’ensemble des technologies de l’informatique, de l’électronique et des télécommunications dans un habitat, représente une solution technologique possible. L’objectif de ce chapitre est d’introduire la domotique comme une solution pertinente pour répondre à ces enjeux. La section 1.1 présente le contexte démographique et technologique : les tendances et les enjeux du vieillissement des populations (intimement lié au phénomène de dépendance) et les solutions technologiques qui peuvent y apporter des réponses. Nous pensons que les systèmes domotiques peuvent dès à présent permettre d’améliorer cette situation (section 1.2). Dans la section 1.3, nous précisons les motivations qui ont permis de guider ces travaux et les contributions scientifiques apportées.

## Sommaire

---

|         |   |    |
|---------|---|----|
| 1.1     | Contexte . . . . .  | 11 |
| 1.1.1   | Évolution démographique . . . . .                         | 11 |
| 1.1.1.1 | Vieillessement des populations . . . . .                  | 11 |
| 1.1.1.2 | Prévalence de la dépendance . . . . .                     | 14 |
| 1.1.2   | Processus de Production du Handicap . . . . .             | 15 |
| 1.1.2.1 | Introduction au modèle conceptuel . . . . .               | 15 |
| 1.1.2.2 | Handicap : une situation universelle . . . . .            | 17 |
| 1.1.2.3 | Aménagement du lieu de vie . . . . .                      | 18 |
| 1.1.3   | Évolution des systèmes domotiques . . . . .               | 20 |
| 1.1.3.1 | Émergence des systèmes pervasifs . . . . .                | 20 |
| 1.1.3.2 | Évolution de la domotique . . . . .                       | 21 |
| 1.2     | La domotique pour l'assistance à la personne . . . . .    | 23 |
| 1.2.1   | Description d'un système domotique d'assistance . . . . . | 24 |
| 1.2.1.1 | Définition(s) . . . . .                                   | 24 |
| 1.2.1.2 | Caractéristiques . . . . .                                | 25 |
| 1.2.2   | État de l'art des projets existants . . . . .             | 27 |
| 1.2.2.1 | Projets d'habitats intelligents . . . . .                 | 27 |
| 1.2.2.2 | Projets du Lab-STICC / MOCS . . . . .                     | 31 |
| 1.2.3   | Verrous scientifiques identifiés . . . . .                | 34 |
| 1.3     | Motivations & Contributions . . . . .                     | 35 |
| 1.3.1   | Motivations . . . . .                                     | 35 |
| 1.3.1.1 | Conception universelle / centrée utilisateur . . . . .    | 35 |
| 1.3.1.2 | Nouveau schéma de collaboration . . . . .                 | 36 |
| 1.3.1.3 | Supervision non-intrusive d'interactions . . . . .        | 37 |
| 1.3.2   | Contributions scientifiques . . . . .                     | 38 |
| 1.3.2.1 | Langage de modélisation domotique . . . . .               | 38 |
| 1.3.2.2 | Flot de conception dirigé par les modèles . . . . .       | 39 |
| 1.3.2.3 | Modèle de supervision basé sur les ontologies . . . . .   | 40 |

---

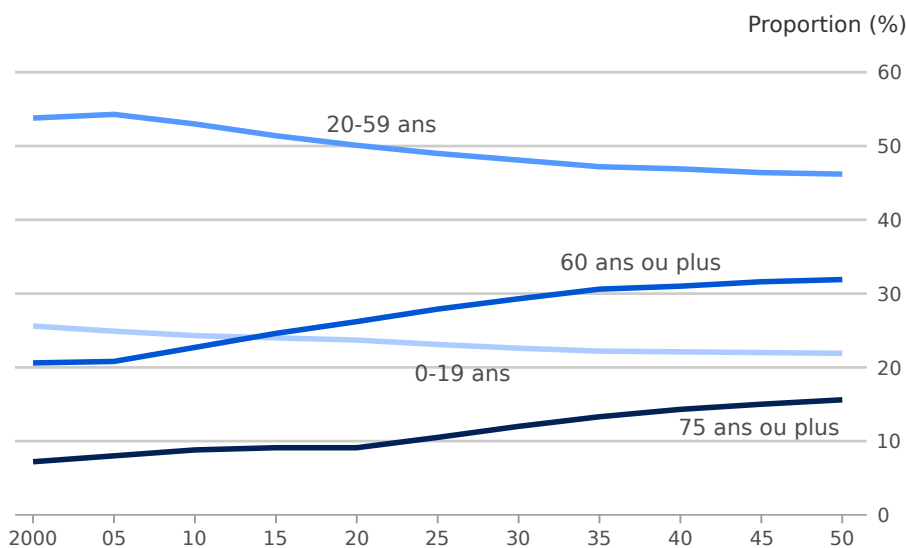
## 1.1 Contexte

Le vieillissement de la population augmente inéluctablement la prévalence de la dépendance et amène parfois à des situations de handicap (section 1.1.1). Dans la section 1.1.2, nous présentons le *Processus de Production du Handicap* qui fournit un cadre universel pour comprendre les situations de handicap et agir en conséquence. Si les déficiences liées à l'âge s'accompagnent parfois de limitations fonctionnelles, les facteurs environnementaux permettent toutefois de les compenser. L'évolution des systèmes domotiques permet d'y apporter des solutions technologiques (section 1.1.3).

### 1.1.1 Évolution démographique

#### 1.1.1.1 Vieillesse des populations

**Tendance à l'échelle nationale** Selon les récentes projections de l'INSEE, la France métropolitaine compterait 70,0 millions d'habitants au 1er janvier 2050, soit 9,3 millions de plus qu'en 2005 [Robert-Bobée 2006]. En supposant que les tendances démographiques récentes se maintiennent, la population augmenterait sur toute la période, mais à un rythme de moins en moins rapide. En 2050, un habitant sur trois serait âgé de 60 ans ou plus (31,9%), contre un sur cinq en 2005 (20,8%) (cf figure 1.1). La part des jeunes diminuerait ainsi que celle des personnes d'âge actif, passant respectivement de 24,9% à 21,9% (0-19 ans) et de 54,3% à 46,2% (20-59 ans) pendant cette période.

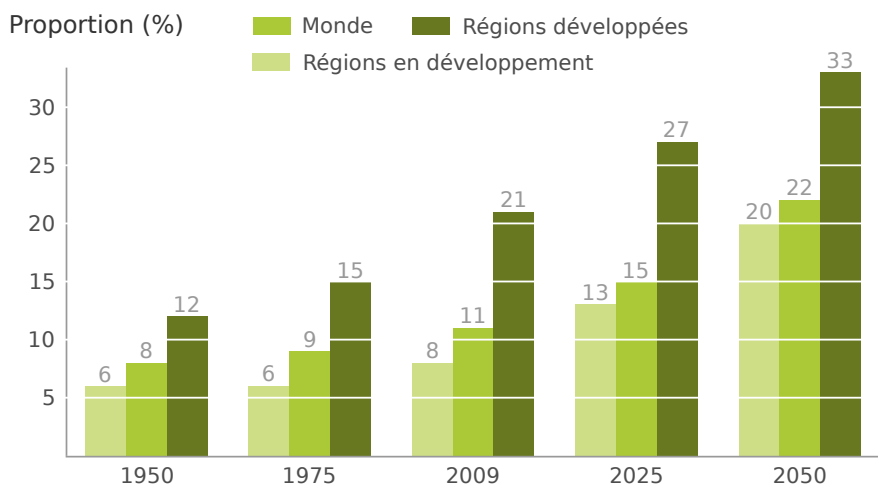


Source : Insee, situations démographiques et projections de population 2005-2050, scénario central.

FIGURE 1.1 – Évolution de la population française métropolitaine de 2000 à 2050

Les résultats présentés sur la figure 1.1 se basent sur l'évolution démographique française passée pour en dégager une tendance : on parle de "scénarios tendanciels". Ces projections se basent principalement sur des hypothèses de fécondité (dont l'indicateur se maintiendrait à 1,9 enfant par femme sur toute la période), de mortalité (qui devrait continuer à baisser selon la tendance estimée sur la période 1988 à 2002) et de solde migratoire (supposé se maintenir à plus de 100 000 personnes par an, niveau moyen des années 2004-2005). Si l'on exclut tout scénario "catastrophe" (e.g. une épidémie de grippe) qui entraînerait le décès d'un grand nombre de personnes âgées, il est possible d'avoir une photographie assez précise de ce que sera la structure de la population française d'ici à quarante ans.

**Tendance à l'échelle mondiale** La tendance observée en France se généralise à l'échelle mondiale, et en particulier dans les pays développés. En ajustant la fenêtre d'observation à un cadre temporel et géographique plus large, on s'aperçoit que le vieillissement des populations est un phénomène inéluctable et international. Le *baby boom* de l'après-guerre est la conséquence immédiate de ce phénomène dans les pays développés. C'est entre 2006 et 2035 que l'on verra l'arrivée à des âges avancés des générations nombreuses nées pendant cette période (entre 1946 et 1975). Entre 2035 et 2050, la hausse serait plus modérée, les personnes qui atteindront 60 ans appartenant à des générations moins nombreuses. De plus, les générations du baby-boom approcheront 90 ans : elles parviendront donc à des âges de forte mortalité. Mais l'allongement de la durée de vie et la baisse de la natalité dans les pays développés permet à ce phénomène démographique de continuer à croître malgré tout.



Source : World Population Ageing 2009, United Nations Department of Economic and Social Affairs.

FIGURE 1.2 – Évolution de la population mondiale (> 60 ans) de 1950 à 2050

Le vieillissement de la population est un phénomène global puisqu'il touche quasiment toutes les régions du monde. La figure 1.2 permet d'illustrer l'universalité du phénomène, tout en montrant les différences entre les régions développées et les régions en développement [United Nations 2009]. Selon l'organisation des *Nations Unies*, les régions en développement incluent toutes les régions d'Afrique, d'Asie (en excluant le Japon), d'Amérique Latine et Caraïbes et d'Océanie (en excluant l'Australie et la Nouvelle-Zélande). Les régions développées représentent donc toutes les autres régions auxquelles viennent s'ajouter les pays exclus des régions en développement. Dans les pays les plus développés, plus d'un cinquième de la population est actuellement âgé de 60 ans ou plus alors qu'en 2050, ce sera près d'un tiers de la population. Dans les pays en développement, les personnes âgées comptabilisées à l'heure actuelle représentent 8% de la population. En 2050, ces pays atteindront une proportion de personnes âgées comparable au taux actuel des pays développés, soit un cinquième de la population.

**Un enjeu sociétal et économique** Le vieillissement des populations est profond, impliquant des conséquences majeures et un impact sur toutes les facettes de la société. D'un point de vue économique, ce phénomène aura un impact direct sur la croissance économique, les épargnes, l'investissement, la consommation, le marché du travail, les retraites, la fiscalité et les transferts intergénérationnels. D'un point de vue social, le vieillissement de la population influence la composition et les habitudes de vie familiale, la demande de logements, les tendances migratoires et le besoin de services d'aide à la personne et de soins de santé.

Sur ce dernier point, les pays développés doivent dès à présent faire face aux problématiques suivantes [Chan *et al.* 2008] :

- Les soins de santé et la prise en charge de personnes dépendantes requièrent une efficacité accrue du fait de la diminution du nombre d'aidants par rapport aux personnes nécessitant une prise en charge [Paré *et al.* 2007].
- Les demandes de soins de santé et le coût associé augmentent considérablement : en France, le montant des dépenses courantes de santé s'élève à 240 milliards d'euros en 2011, soit 12% du PIB [INSEE 2012]. Aux États-Unis, cela représente la section des dépenses publiques la plus importante [Schlender 2003].
- Les soins de santé à distance ainsi que les soins à domicile sont de plus en plus demandés pour éviter les hospitalisations à long terme [Bodenheimer *et al.* 2002]. De nombreuses recherches ont été menées dans le domaine de l'informatique médicale et des différents domaines associés pour développer des technologies fiables et adaptées pour supporter ce changement [Haux 2006].
- Les maladies et handicaps sont en constante augmentation : l'association *US Alzheimer Association* a calculé une augmentation annuelle du coût causé par les démences de type Alzheimer, passé de 33 milliards de dollars en 1998 à 61 milliards en 2002 [Koppel 2002].

### 1.1.1.2 Prévalence de la dépendance

**Vieillesse et dépendance** La dépendance se définit comme l'impossibilité partielle ou totale pour une personne d'effectuer sans aide les activités de la vie quotidienne et de s'adapter à son environnement. L'âge en est le plus grand facteur de risques relatif aux soins de santé et à l'aide à la personne [Dupré & Karjalainen 2003]. La figure 1.3 met l'accent sur les problèmes de santé de longue durée et les déficiences<sup>1</sup> selon l'âge dans 24 pays européens. Le taux de personnes dépendantes âgées de 16 à 64 ans augmente linéairement avec l'âge. De 6,9% dans la tranche d'âge 16-24 ans, il passe à 14,6% chez les 40-44 ans, pour atteindre 30,7% chez les 60-64 ans.

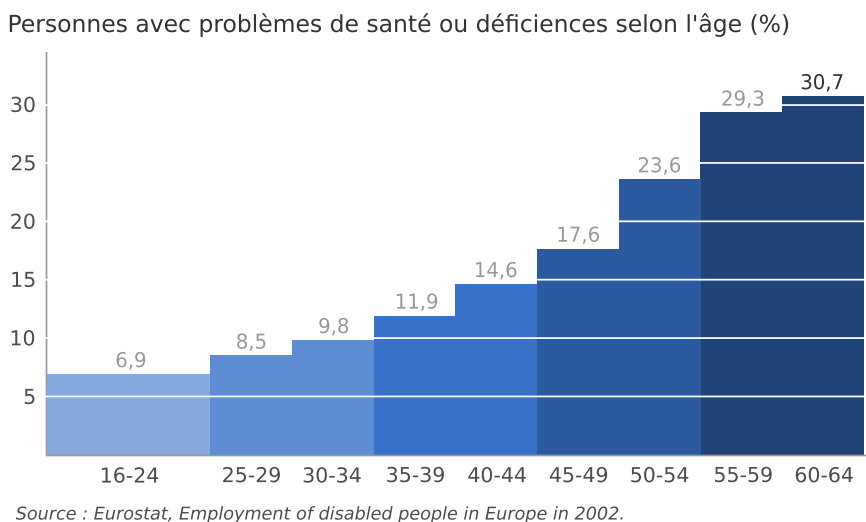


FIGURE 1.3 – Personnes avec problèmes de santé ou déficiences selon l'âge en Europe

**Causes de la dépendance chez les seniors** La dépendance des personnes âgées peut être attribuée à différentes causes. Les maladies représentent en premier lieu une cause importante de survenue ou d'aggravation de la dépendance. Les affections dégénératives du système nerveux central telles que les démences, les maladies destructrices des articulations (coxarthrose, gonarthrose), les accidents vasculaires cérébraux et les maladies chroniques (diabète, cancers, troubles de la santé mentale, etc.) en sont des exemples. La baisse des accidents mortels et l'augmentation relative des traumatismes physiques non-mortels chez les seniors [Bauer & Steiner 2009] expliquent aussi l'augmentation du nombre de personnes dépendantes. Les chutes, témoins d'une baisse inéluctable des capacités physiques, en représentent une cause majeure : si chaque an-

1. *Long-Standing Health Problem or Disability (LSHPD)* : problèmes de santé de longue durée ou déficiences (altérations corporelles ou dysfonctionnement des diverses parties du corps ou du cerveau) pendant 6 mois ou plus, ou qui devraient durer 6 mois ou plus [Dupré & Karjalainen 2003]. L'enquête porte sur des habitants de logements privés individuels.

née, 120000 personnes de l'Union Européenne âgées de 60 ans ou plus meurent des conséquences de traumatismes physiques, bien plus de personnes qui ont subi un tel traumatisme deviennent par la suite dépendantes.

Derrière les statistiques présentés dans cette section, il ne faut pas oublier les particularités et les besoins spécifiques de chaque individu : il n'existe pas de liens mécaniques entre les déficiences liées à l'âge et le handicap. Un certain nombre de personnes âgées qui vivent avec une ou des déficiences ne se déclarent pas et ne sont pas reconnues comme étant en situation de handicap. L'entourage familial, l'environnement social et culturel, la propre perception de la personne influent sur le fait de la ressentir ou de la faire reconnaître [Betremieux *et al.* 2010]. Mais plus le nombre de déficiences augmente, plus ces déficiences s'accompagnent de limitations fonctionnelles et donc de situations de handicap. Parmi les personnes âgées qui sont en situation de handicap, on peut distinguer les personnes qui l'étaient avant le processus de vieillissement de celles qui ne l'étaient pas.

## 1.1.2 Processus de Production du Handicap

### 1.1.2.1 Introduction au modèle conceptuel

Les personnes en situation de handicap, que cette situation soit liée à l'âge et/ou au handicap lui-même, représentent environ 15% de la population mondiale. D'après le premier *Rapport mondial sur le handicap*<sup>2</sup> [World Health Organization 2011], plus d'un milliard de personnes sont atteintes d'une forme ou d'une autre de handicap sur notre planète. Parmi ces personnes, près d'une sur cinq a de très grandes difficultés fonctionnelles ou souffre d'un handicap dit "sévère". Le *Processus de Production du Handicap* (PPH) fournit un cadre formel pour comprendre les situations de handicap et agir en conséquence. Il s'agit d'un modèle conceptuel issu des travaux du Comité de révision de la proposition québécoise de classification [Fougeyrollas & RIPPH 1998] qui promeut une approche globale du handicap où la personne est considérée en relation permanente avec son environnement physique et social. Le PPH prend en considération les dynamiques entre les éléments favorisant ou entravant la pleine participation sociale de la personne. La participation sociale de tout individu est ainsi vue comme le résultat de l'interaction permanente entre les *facteurs personnels* (intrinsèques) et les *facteurs environnementaux* (extrinsèques).

À partir de ce modèle simple mais fondamental, le PPH propose de mettre en évidence le phénomène de production de différences ou de distinctions en introduisant les causes et conséquences des maladies et traumatismes. Il permet de mettre en évidence la variation de possibilités en relation avec la norme biologique, fonctionnelle et sociale dont il tente de clarifier le processus de production. La figure 1.4 illustre ce processus et les différents éléments le constituant :

---

2. Rapport issu d'un travail collaboratif entre l'Organisation Mondiale de la Santé et la Banque mondiale



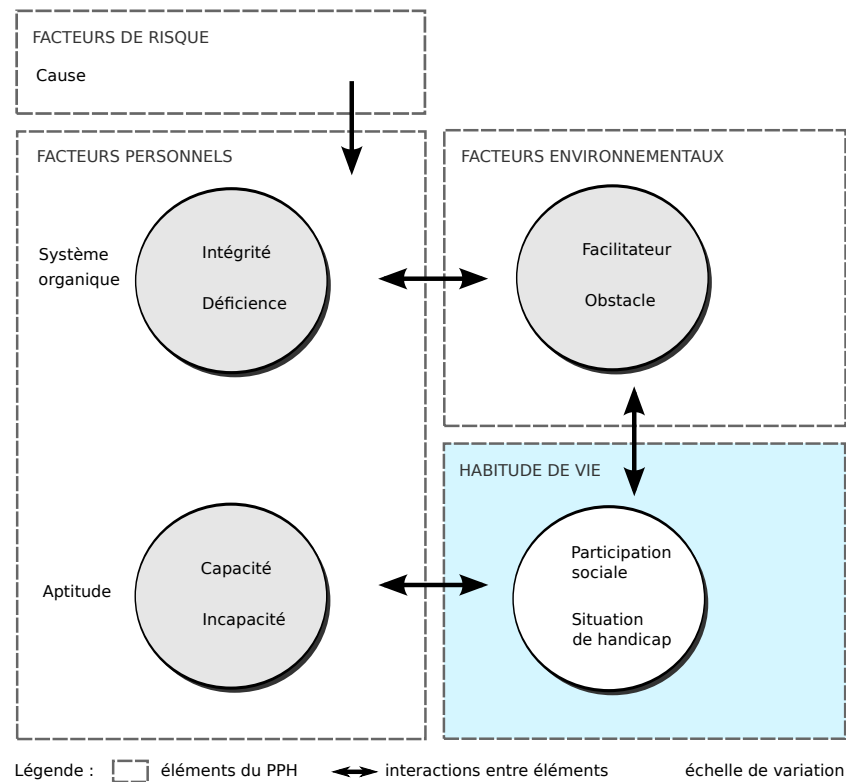


FIGURE 1.4 – Schéma conceptuel du Processus de Production du Handicap

- Les *facteurs de risques* appartiennent à l’individu ou à l’environnement. Ils sont susceptibles d’altérer le développement de la personne en provoquant une maladie, un traumatisme ou toute autre atteinte à l’intégrité.
- Les *facteurs personnels* sont les caractéristiques propres à la personne (âge, sexe, identité culturelle, etc.). Entre autres, on y distingue l’état organique et les aptitudes. Une *déficience* est relative au degré d’atteinte anatomique, histologique<sup>3</sup> ou physiologique d’un système organique. Une *incapacité* est relative au degré de réduction d’une aptitude (possibilité pour une personne d’accomplir une activité physique ou mentale sans tenir compte de l’environnement).
- Les *facteurs environnementaux* sont les éléments représentatifs des dimensions sociales ou physiques qui déterminent l’organisation et le contexte d’une société. Ces éléments peuvent être les attitudes de l’employeur et des collègues, l’accès aux aides techniques, l’accessibilité du bâtiment, la disponibilité d’une adaptation à l’organisation des tâches, la prise en compte des variables climatiques, l’organisation des services de formation professionnelle ou de réadaptation profes-

3. L’histologie est la branche de la biologie et de la médecine qui étudie les tissus biologiques. (source : Wikipédia)

sionnelle, les mesures d'égalité des chances dans les lois liées au travail et, le plus important, la disponibilité d'un emploi sur le marché du travail.

- Les *habitudes de vie* sont définies comme des activités quotidiennes ou des rôles sociaux valorisés par le contexte socioculturel pour une personne. Elles assurent la vie et l'épanouissement dans une société ou au contraire engendrent une situation de handicap totale.

### 1.1.2.2 Handicap : une situation universelle

L'introduction de ce modèle anthropologique de développement humain permet de voir le handicap comme un état non figé mais évolutif, s'inscrivant dans une perspective universelle. Il n'y a pas deux mondes, celui des bien portant et celui des handicapés, mais un seul où le phénomène des déficiences et des incapacités n'en est qu'une modalité, qu'une variation. La situation de handicap est ainsi considérée comme variable en fonction du temps, mais aussi en fonction du contexte et de l'environnement. L'uniformité du cadre conceptuel du PPH et les éléments distincts et variables le constituant définissent le handicap comme "une situation singulière qui se conjugue au pluriel" [Ameisen *et al.* 2011]. En effet, le développement de tout individu est conforme au modèle universel du PPH : les déficiences et les incapacités n'en sont que les éléments constitutifs variant dans le temps. Les déficiences, par leur nature ou par leur cumul chez une même personne, peuvent conduire à des limitations dans les actes de la vie quotidienne, y compris les plus simples comme marcher, se baisser ou parler. Pour autant, les restrictions d'activités occasionnées ne conduisent pas automatiquement à des situations de handicap. Deux raisons peuvent expliquer cela :

1. Ces restrictions peuvent être plus ou moins fortes : pour les troubles visuels, les degrés de sévérité peuvent par exemple s'échelonner de la myopie à la malvoyance voire la cécité.
2. Le contexte social dans lequel elles s'exercent joue un rôle primordial : une même déficience peut être vécue de manière très différente d'une personne à l'autre selon son cadre de vie, son entourage familial ou son milieu socioprofessionnel.

#### DEFINITION DU HANDICAP

"Constitue un handicap, au sens de la présente loi, toute limitation d'activité ou restriction de participation à la vie en société subie dans son environnement par une personne en raison d'une altération substantielle, durable ou définitive d'une ou plusieurs fonctions physiques, sensorielles, mentales, cognitives ou psychiques, d'un polyhandicap ou d'un trouble de santé invalidant."

Loi du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées [Chossy 2006]

La définition de la loi française du 11 février 2005 dévoile la pluralité des situations de handicap, intimement liées aux facteurs personnels et environnementaux. Elle dépasse l'ancien cadre institutionnel qui limitait le champ aux allocataires de prestations ou aux personnes ayant une reconnaissance officielle d'un handicap. Le ressenti des individus ainsi que leurs relations avec leur environnement sont maintenant parties prenantes de cette définition. Une personne peut être considérée en situation de handicap si elle s'inscrit dans au moins une des trois approches complémentaires proposées dans [INSEE 2010a, INSEE 2010b] :

1. Une personne au *handicap reconnu* : déclare avoir une ou des reconnaissances administratives telles que des allocations, des pensions d'invalidité, des rentes d'incapacité ou une reconnaissance de la part de la maison départementale des personnes handicapées (MDPH)
2. Une personne au *handicap identifié* : déclare une impossibilité totale à effectuer au moins une tâche courante (voir, parler, marcher, se souvenir, etc.) ou recevoir une aide d'un tiers en raison de son handicap, un aménagement spécial de son logement ou l'utilisation de toute aide technique (e.g. prothèse, appareillage)
3. Une personne au *handicap ressenti* : déclare considérer avoir un handicap

Une situation de handicap peut finalement être considérée comme un état qui peut être modifié grâce à la réduction des déficiences, le développement des aptitudes et l'adaptation de l'environnement. (Par exemple, certaines personnes âgées dépendantes qui vivent au deuxième étage d'un immeuble peuvent se retrouver en situation de handicap si l'ascenseur ne fonctionne plus.) L'adaptation de l'environnement fait référence dans un premier temps à des adaptations sur le plan social (e.g. attitudes de l'employeur et des collègues, la disponibilité d'un travail sur le marché de l'emploi, la présence et le soutien de la famille). Dans un deuxième temps, il fait référence à l'adaptation physique de l'environnement (e.g. accès aux aides techniques, accessibilité du bâtiment). Les aides techniques pour compenser les limitations physiques et les complications associées répondent à des besoins très variés. On y retrouve, entre autres, les aides à la marche (canne, déambulateur, etc.), aux déplacements (fauteuil roulant, tricycle, etc.), aux transferts (planche, sangle, lève-personnes, etc.) ou encore des prothèses extérieures (prothèses des bras ou des jambes). Les aides techniques présentes à domicile et les aménagements du logement jouent également un rôle important et peuvent bénéficier au plus grand nombre. Elles recouvrent un large éventail de possibilités répondant ainsi à des besoins variés et à des handicaps différents.

### 1.1.2.3 Aménagement du lieu de vie

L'accessibilité et l'aménagement du lieu de vie comprend un grand nombre de possibilités d'adaptation suivant les besoins de ses occupants [Espagnacq 2012] :

1. *Les aménagements architecturaux* comme les ouvertures de portes et les couloirs spécialement élargis, facilitent le passage d'un fauteuil roulant. Le regroupement

de pièces, en particulier les toilettes et la salle de bain, agrandit l'espace et offre par exemple la possibilité d'y installer un fauteuil de douche.

2. Les *meubles adaptés* : les toilettes, les tables, les sièges, les lits, les placards, etc. L'abaissement du plan de travail de la cuisine permet par exemple à une personne en fauteuil roulant de se préparer seule les repas, si toutefois cette dernière en a les capacités gestuelles.
3. Les *dispositifs de soutien et d'aide aux transferts* : des barres d'appui dans la salle de bain permettent par exemple de sortir plus facilement de la baignoire ou des wc. Il en est de même pour les mains courantes et rampes installées au niveau des marches d'escalier pour faciliter le passage d'un étage à un autre.

La norme *NF P91-201* définit un cadre légal reprenant la plupart de ces adaptations. Elle définit les conditions générales d'accessibilité des logements et autres bâtiments et de leurs équipements pour les personnes en situation de handicap physique<sup>4</sup>. On y retrouve notamment les dimensions à respecter pour la largeur des couloirs, la superficie des pièces, les aires de dégagement permettant le changement de direction d'un fauteuil, etc. Si le respect de ces préconisations ne permet pas d'adapter entièrement le logement à une personne, il permet malgré tout de définir une base à partir de laquelle des adaptations à moindre frais pourront être réalisées.

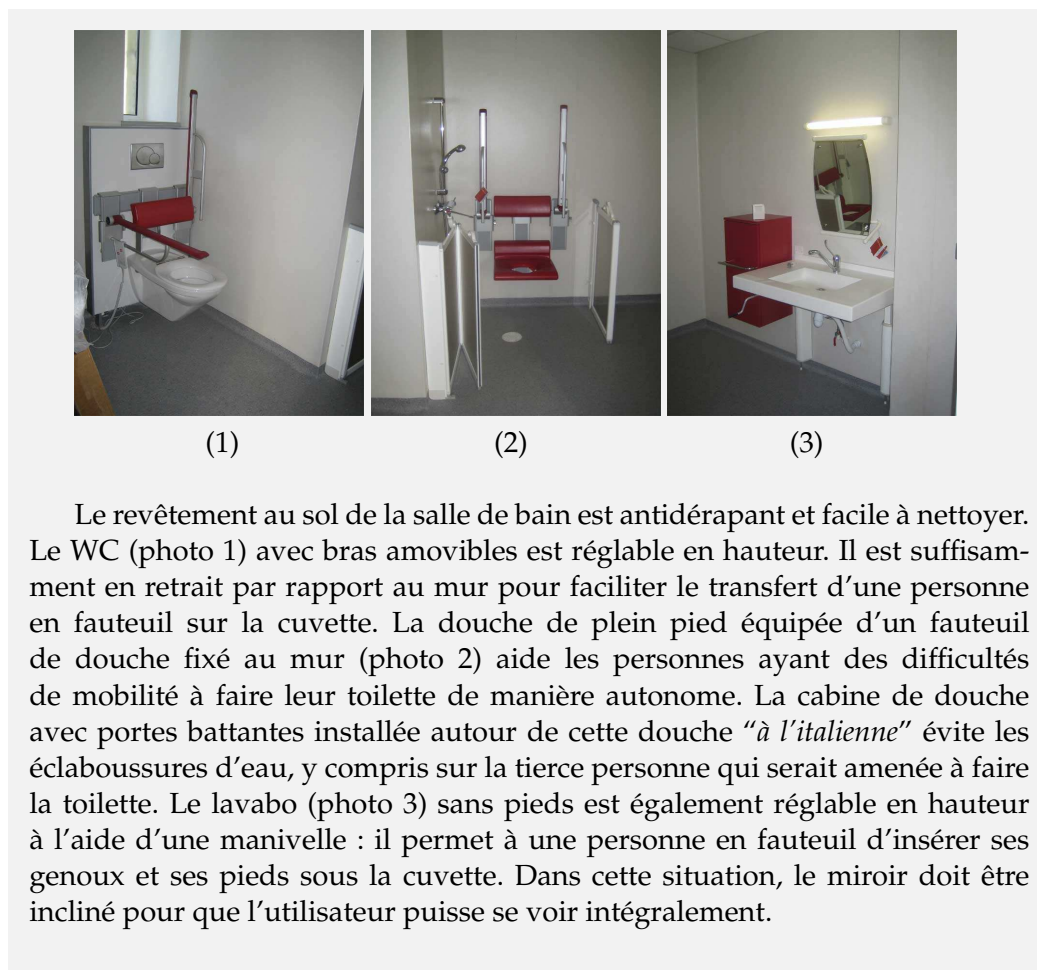
### LES APPARTEMENTS TREMPLINS DU CMRRF DE KERPAPPE

Les appartements tremplins du CMRRF (*Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles*) de Kerpappe sont conçus pour évaluer l'aptitude au retour à domicile de personnes en situation de handicap suite à un accident ou à une maladie invalidante. La figure ci-dessous illustre les différents équipements et aménagements réalisés dans la salle de bain d'un de ces appartements.



---

4. Ces spécifications pourraient à fortiori être étendues aux cas des handicapés semi-ambulants et à celui des personnes âgées ayant des difficultés d'atteinte et de préhension. Elles conviennent également aux personnes non-handicapées sans gêne notable [Habitat Senior 2012].



Les aménagements architecturaux, les équipements et mobiliers adaptés ou encore les dispositifs de soutien ou d'aide aux transferts ne sont pas les seules adaptations que l'on retrouve dans un logement. Ces dernières peuvent être couplées avec l'utilisation de l'électronique, de l'informatique ou encore des télécommunications pour permettre le contrôle à distance et l'automatisation de l'habitat : on parle de domotique. Ce domaine d'application bénéficie de l'évolution des systèmes électroniques pour répondre au mieux aux besoins de personnes dépendantes.

### 1.1.3 Évolution des systèmes domotiques

#### 1.1.3.1 Émergence des systèmes pervasifs

La course à la miniaturisation des équipements électroniques, l'amélioration de leur capacités de calcul et de télécommunication bouleversent l'utilisation des technologies de l'information et de la communication. Les ordinateurs individuels (*PC : Personal*

*Computer*) de bureau laissent place aux tablettes, *netbook*, *smart phones* que l'utilisateur peut utiliser pour contrôler son environnement, peu importe où il se trouve.

La figure 1.5 (couleur grise) reprend les grandes étapes de l'évolution des systèmes électroniques, du PC de bureau aux équipements électroniques miniaturisés et distribués dans l'environnement [Waldner 2007]. Après la naissance de l'informatique, le PC fixe de bureau a rapidement émergé, permettant à un utilisateur de manipuler une Interface Home-Machine (IHM) unique pour contrôler ce dispositif dans un but précis. L'amélioration des capacités de télécommunication et de mise en réseaux avec notamment l'introduction d'Internet permet l'interconnexion de ces PC. Un utilisateur peut communiquer et échanger avec un autre utilisateur par l'intermédiaire de ces ordinateurs individuels. L'amélioration des capacités de stockage d'énergie (i.e. utilisation de batteries) rend l'utilisation de ces dispositifs mobiles, portables. Les briques élémentaires des systèmes électroniques étant posées, on assiste à une multiplication et une miniaturisation de ces systèmes, qualifiés alors de pervasifs : c'est la naissance de l'informatique ubiquitaire que Mark Weiser avait déjà en tête près de 10 ans auparavant [Weiser 1991]. Par opposition au paradigme du PC, l'informatique ubiquitaire voit l'utilisateur interagir avec un plus grand nombre de dispositifs et de systèmes informatiques simultanément au cours de ses activités journalières, sans qu'il en soit conscient pour autant. La distribution de ces dispositifs miniatures amène à penser à une distribution de l'intelligence où à chacun de ces objets est associé un agent logiciel avec ses propres connaissances et ses capacités de décision. La coordination de ces agents logiciels fait émerger ce que l'on appelle de l'"intelligence ambiante" visant à assister l'utilisateur dans ses activités de la vie quotidienne.

### 1.1.3.2 Évolution de la domotique

#### DEFINITION DE LA DOMOTIQUE (1)

Étymologiquement, le terme domotique vient du latin "*domus*" qui signifie *maison* associé au suffixe "*-tique*" dérivé de *automatique*. Le contrôle des éclairages, la sécurité et la gestion de l'énergie sont autant d'exemples d'application qui requièrent la coordination d'une multitude d'équipements de nature hétérogène pour assurer un plus grand confort, optimiser la sécurité ou encore diminuer la consommation énergétique.

La domotique fait référence à l'ensemble des équipements électroniques et informatiques que l'on retrouve dans un habitat pour coordonner et automatiser son fonctionnement. De part sa nature, la domotique s'inscrit dans le paradigme des systèmes pervasifs : elle a donc également suivi les étapes de l'évolution des systèmes électroniques. La figure 1.5 (couleur bleue) propose de faire le parallèle avec l'évolution des systèmes électroniques introduite dans [Waldner 2007]. Elle reprend les grandes étapes

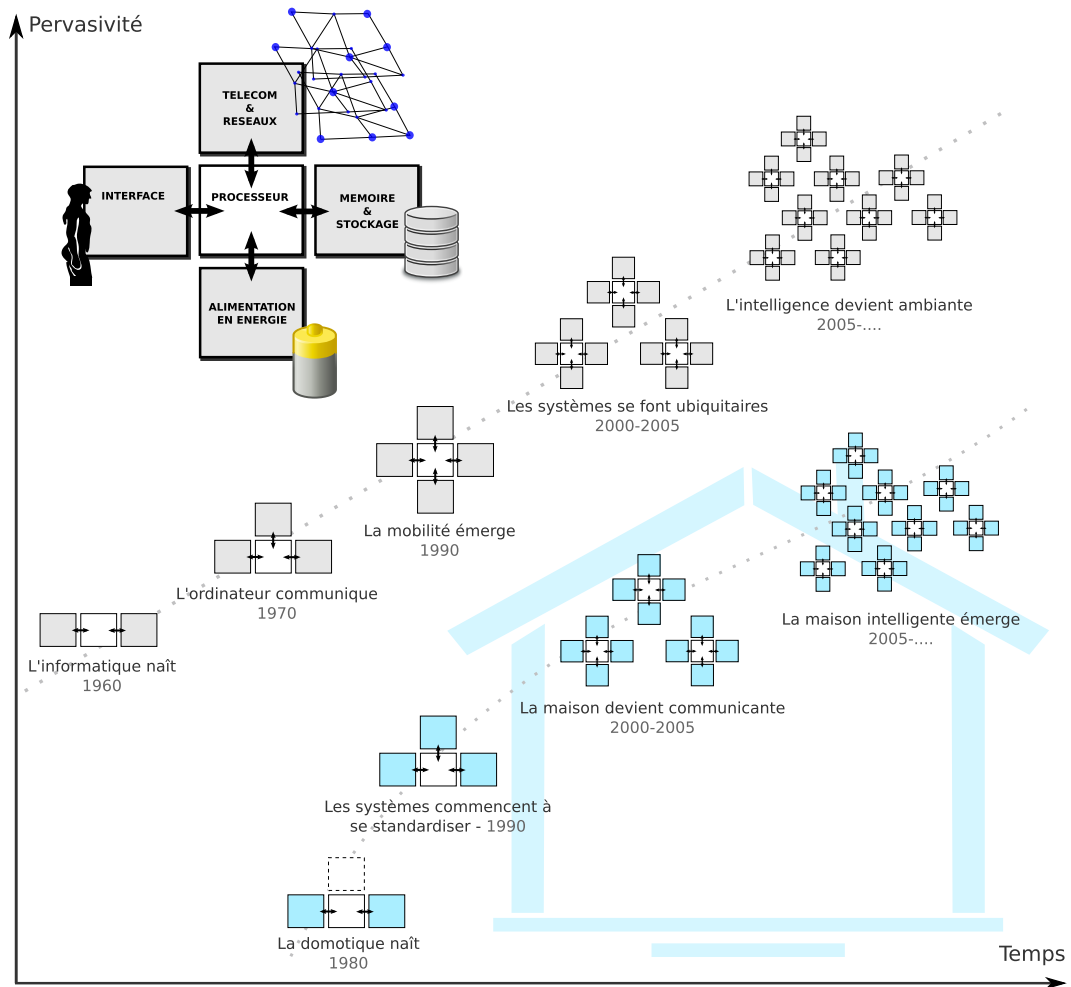


FIGURE 1.5 – Évolution des systèmes électroniques et des systèmes domotiques

de l'évolution de la domotique, depuis sa naissance jusqu'à sa dernière évolution où l'"intelligence" est répartie dans les équipements électroniques distribués dans l'habitat. Les premières applications domotiques apparaissent au début des années 80 avec l'introduction de l'infrarouge pour le contrôle à distance et le protocole X10 (créé en 1978) pour commander les équipements domestiques par courant porteur. Dans un premier temps, ce protocole qui utilise le réseau électrique existant répond principalement aux besoins de sécurité et de confort : gestion des lumières, du chauffage, de l'alarme, etc. *Home Minder* est un exemple de système utilisant les technologies infrarouge et courant porteur (X10) [Brody 1985] offrant ces fonctionnalités. Commercialisé par *General Electric*, c'est un des premiers systèmes qui propose une interface graphique affichée sur l'écran de la télévision. Les années 90 sont marquées par l'apparition de standards de communication par bus domotique dont le but "utopiste" est de garantir l'interopéra-

bilité<sup>5</sup> des équipements domotiques de natures hétérogènes (provenant de différents équipementiers). *KNX-EIB* et *LonWorks-LON* sont deux normes de standardisation issues d'un consensus (des acteurs du domaine) qui se sont imposées respectivement en Europe et aux États-Unis. La technologie *KNX* est ouverte et interopérable avec la possibilité d'intégrer de nombreux modules de transmission : paires torsadées, radio, courant porteur, ethernet, infrarouge, etc. Malgré ces avancées technologiques qui offrent de nouvelles possibilités, la domotique ne trouve pas sa place sur le marché, faute d'un besoin mal identifié. Les installations domotiques sont globalement vues comme trop "gadgets" [PIPAME 2009], trop coûteuses et difficiles à maîtriser. Pour démocratiser l'utilisation de la domotique, on met en avant au début des années 2000 les services que peut proposer la maison en dissimulant les systèmes électroniques et les technologies sous-jacentes. Les interactions entre l'utilisateur et sa maison sont ainsi mises au premier plan. L'interconnexion des équipements permet de regrouper le contrôle à distance, la gestion d'énergie et toutes les autres fonctionnalités domotiques : on parle de *maison communicante*. Un détecteur de présence peut par exemple servir pour l'allumage automatique des lumières suivant la localisation de la personne mais il peut aussi être utilisé en tant qu'alarme sous une certaine plage horaire. Les technologies sans fils radio comme le *Zigbee* ou le *Z-Wave* participent à cette évolution. Elles sont basées sur un réseau de modules de communication (modules discrets qui consomment peu) dont la structure du réseau peut être personnalisée pour améliorer les transmissions (amélioration des distances par rapport à la couverture radio). Le paradigme de *maison intelligente* prône l'utilisation de tous ces équipements diffus et interconnectés pour automatiser le fonctionnement de l'habitat et s'adapter automatiquement aux besoins de ses occupants. La multitude de capteurs présents dans le logement permet d'enregistrer un grand nombre d'informations sur son utilisateur pour par la suite lui proposer des services contextuels et adaptés à ses besoins. Bien que cette appellation ait été entendue depuis des années, l'objectif de rendre la maison "intelligente" est toujours d'actualité, notamment pour permettre l'assistance aux personnes dépendantes.

## 1.2 La domotique pour l'assistance à la personne

La domotique pour l'assistance à la personne<sup>6</sup> représente une déclinaison possible des systèmes pervasifs que l'on retrouve dans l'habitat (section 1.2.1). Témoins d'un véritable engouement pour ce domaine d'application, de nombreux projets d'"habitats intelligents" ont vu le jour ces dernières années (section 1.2.2). Il reste malgré tout des verrous à lever pour améliorer l'adaptation et l'automatisation de l'habitat : les problématiques liées à la conception de ces systèmes sont présentées dans la section 1.2.3.

---

5. Il s'agit ici d'une étape de l'évolution des systèmes domotiques qui marque la volonté d'utiliser des normes de standardisation adoptées par un consortium d'entreprises pour que des équipements issus de différents fournisseurs puissent communiquer entre eux.

6. On retrouve souvent le terme *Ambient Assisted Living* pour faire référence à ce type de systèmes.



## 1.2.1 Description d'un système domotique d'assistance

### 1.2.1.1 Définition(s)

Les technologies évoluant en permanence, il n'existe pas de définition consensuelle des systèmes domotiques d'assistance, communément appelés "*habitats intelligents*". Les besoins de l'utilisateur permettent de décliner ce concept en différents systèmes d'assistance vus sous un angle différent suivant les applications qui y sont développées. Malgré tout, ces systèmes reposent sur un socle commun.

#### DEFINITION DE LA DOMOTIQUE (2)

"On regroupe sous l'appellation domotique l'ensemble des technologies de l'électronique, de l'informatique et des télécommunications qui sont utilisées dans les domiciles pour rendre ceux-ci plus "intelligents". La domotique vise donc à intégrer dans un tout cohérent différents systèmes intégrant des fonctions de sécurité, de confort, de gestion d'énergie, de communication, de divertissement, d'éducation, etc. qu'on retrouve dans une maison."

Département de génie électrique et de génie informatique, Université de Sherbrooke (Québec)

Pour compléter ou préciser ce qu'on appelle système domotique d'assistance, il faut placer l'utilisateur au centre de cette définition. La domotique permet d'apporter une assistance technique dans l'habitat pour compenser une situation de handicap, permettre le monitoring d'activités ou de données bio-médicales. Selon les besoins de l'utilisateur, différentes déclinaisons de ce même concept peuvent être définies. La littérature [Bonhomme 2008, Chan *et al.* 2008, Chan *et al.* 2009, Noury *et al.* 2003] renvoie aux termes "*technologies d'assistance*" (*assistive technology*), "*télémédecine*" (*telemedicine*), "*e-santé*" (*e-health*), "*télé-santé*" (*tele-health*) ou encore "*gérontechnologie*" (*gerontechnology*). Les applications liées au bio-médical ont tendance à considérer un système domotique comme un moyen de faire de la télémédecine, e-santé, télé-santé, etc. Pour les applications destinées aux personnes âgées dépendantes, on parle plutôt de gérontechnologie. Les applications visant un public de personnes en situation de handicap plus large soulignent quant à elles l'utilisation de technologies d'assistance.

La *télémédecine* (*telemedicine*) fait référence aux systèmes qui exploitent la vidéo, l'audio et d'autres moyens de télécommunications pour transmettre les informations et les données se rapportant aux diagnostics et aux traitements médicaux. Ils fournissent des services de soins et de santé à des patients situés dans des environnements physiques distants. La télémédecine touche entre autres à l'éducation du patient, la prévention de la maladie et la prise de décisions thérapeutiques.

Depuis l'avènement d'Internet et au vu de l'ampleur du multimédia, la télémédecine évolue vers la *télesanté* (*telehealth*). Ce terme semble aujourd'hui plus approprié dans

la mesure où il cadre l'ensemble des technologies, des réseaux et des services de soins basés sur la télécommunication. Il permet d'élargir le champ d'applications issues de la télémédecine en prenant en compte les ressources administratives, les données physiologiques du patient et les bases de données médicales. La télésanté comprend également des programmes d'éducation, de recherche collaborative, de consultation ainsi que d'autres services offerts dans le but d'améliorer la santé du patient [Moore 1999].

Le terme *télésanté à domicile (home telehealth)* agrandit le champ d'applications des télécommunications en permettant de mettre en relation les patients vivants à domicile avec des services de soins, des sources d'informations de santé, d'éducation, etc. Les *soins de santé à domicile (home healthcare)* se sont développés par la suite. Ils définissent les services de soins de santé ainsi que les services sociaux associés fournis aux individus et à leur famille. La *télémédecine à domicile (telehomecare)* rassemble les technologies de télécommunication et de visioconférence pour permettre aux prestataires de santé de communiquer avec les patients lorsqu'ils sont dans leur habitat. On parle de "*visite virtuelle*" qui nécessite l'acquisition (présence de capteurs) et le transfert de données physiologiques du patient (respiration, rythme cardiaque, pression artérielle, etc.) [Boulos *et al.* 2007].

Des *dispositifs d'assistance (assistive devices)* sont conçus et mis en place sur le marché pour accomplir des fonctions personnalisées suivant le besoin de l'utilisateur. Un *système entièrement intégré (integrated system)* est un dispositif dont les fonctions sont commandées à l'aide d'une IHM [Nisbet 1996]. D'autres termes qui renvoient à l'intégration d'équipements électroniques coopérants pour former un système sont introduits : *système domotique intégré (integrated home system)*, *maison intelligente (smart home)* ou encore plus récemment *habitat communicant* [Jeuland 2009]. Les *systèmes intégrés pour la réhabilitation (rehabilitation integrated system)* se rapportent aux dispositifs d'assistance destinés à la réhabilitation des personnes handicapées. Ces dispositifs, la plupart du temps électroniques, se caractérisent par une ergonomie facilitant le contrôle du système. Le terme *maison automatisée (automated house)* se rapporte au concept de *maison intelligente (smart home)* définissant un environnement de vie instrumenté pour aider les personnes à réaliser leurs activités quotidiennes et améliorer leur qualité de vie. L'*immotique* représente enfin la domotique à l'échelle d'un bâtiment, d'un immeuble ou d'un grand site industriel ou tertiaire. Cette échelle implique des solutions techniques de domotique visant à gérer un nombre de modules plus important que pour un simple domicile de particulier.

### 1.2.1.2 Caractéristiques

Malgré le nombre important de déclinaisons du concept de systèmes domotiques d'assistance, on y retrouve malgré tout des modules communs. La figure 1.6 montre l'organisation générale d'un système domotique d'assistance et des modules génériques le constituant.

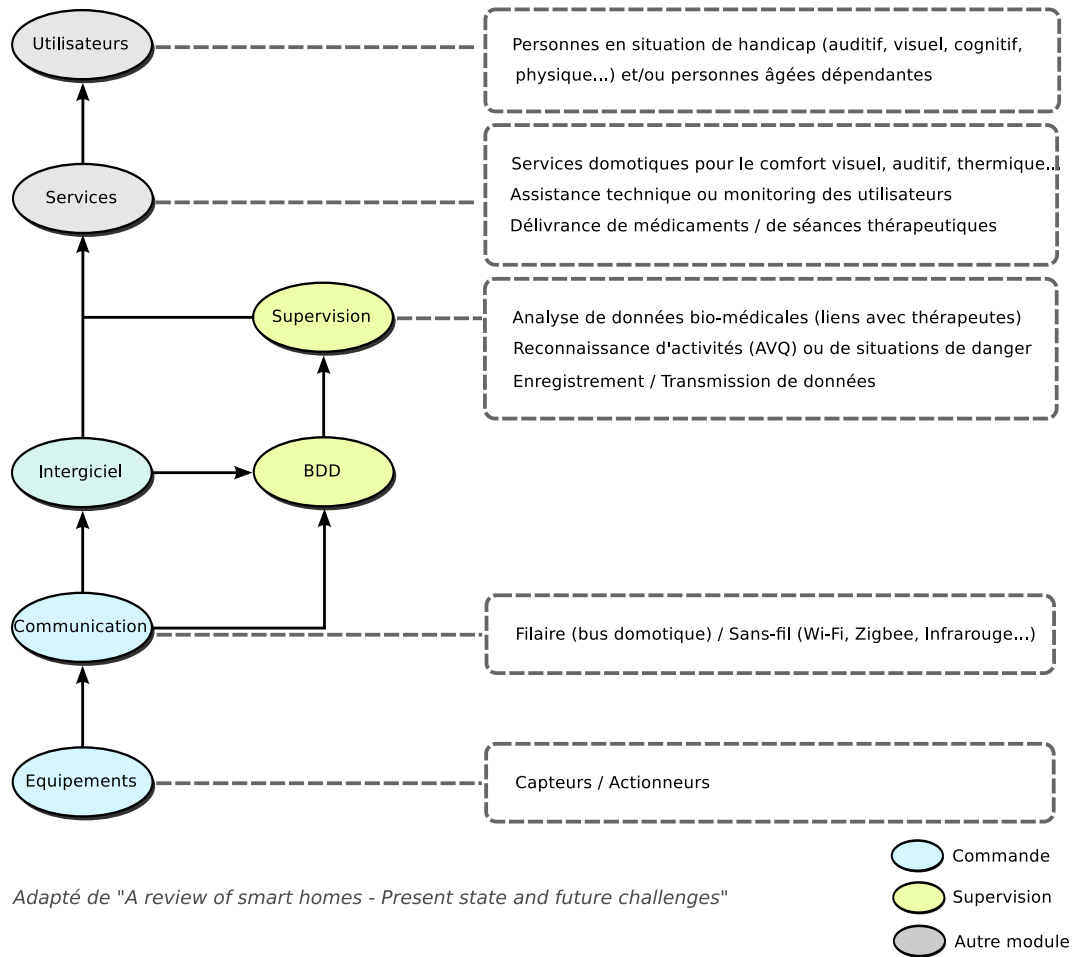


FIGURE 1.6 – Organisation générale d'un système domotique d'assistance

Les *équipements* de type capteurs ou actionneurs sont à la base de tout système domotique pour leur contrôle et leur commande. Un module de *communication* permet ensuite de véhiculer les informations issues de ces équipements pour en garantir l'interopérabilité. Le mode de transmission peut être filaire en utilisant par exemple un bus domotique (*KNX/EIB, Legrand, LonWorks, etc.*), ou bien sans fils via les protocoles *Bluetooth, Wi-Fi, Zigbee, Infrarouge, etc.* L'*intergiciel* (*middleware*) sert d'interface unique entre les couches basses de communication et l'utilisateur. Pour ce dernier, la gestion des différents protocoles de communication est cachée. Une *Base De Données* permet de stocker les informations de capteurs ou d'actionneurs pour alimenter un module de *Supervision*. L'objectif peut être par exemple de faire de la reconnaissance d'Activités de Vie Quotidienne (AVQ). Les *services* sont construits soit au dessus de ce module de supervision (après traitement de l'information), soit après le module intergiciel. L'*utilisateur* dépendant (liée au handicap ou à l'âge) peut enfin bénéficier de ces services

qui peuvent prendre des formes variées suivant le système considéré. Il peut s'agir de la proposition de services en fonction des habitudes de l'utilisateur, ou encore de la gestion d'alerte dans le cas d'une détection d'une situation d'urgence.

Selon les caractéristiques retenues et développées, plusieurs types de systèmes peuvent être identifiés :

1. Systèmes avec des équipements mobiles ou fixes (e.g. capteurs, actionneurs) distribués dans l'habitat pour l'automatisation et/ou le contrôle à distance
2. Systèmes portables (portés par la personne ou implantés)
3. Systèmes sensibles au contexte avec proposition contextuelle et dynamique de services (*context-awareness* [Bricon-Souf & Newman 2007])
4. Systèmes de reconnaissance automatique d'activités (sans intervention humaine)
5. Systèmes d'information pour la collecte et la transmission d'informations relatives aux données biomédicales
6. Systèmes d'assistance robotisée pour l'aide à l'accomplissement de tâches de la vie quotidienne

Dans le cadre de ces travaux, nous nous intéressons aux systèmes domotiques pour l'assistance à la personne de type 1, 3 et 4 que l'on retrouve dans les "*habitats intelligents*". Ce que nous appelons système domotique d'assistance dans la suite de ce mémoire comprend donc des équipements mobiles ou fixes distribués dans l'habitat pour l'automatisation et le contrôle à distance. L'application domotique doit être sensible au contexte pour pouvoir s'adapter aux besoins de l'utilisateur et lui proposer des services à l'exécution. La reconnaissance automatique d'activités de la personne joue également un rôle important pour l'adaptation de services.

## 1.2.2 État de l'art des projets existants

### 1.2.2.1 Projets d'habitats intelligents

Cet état de l'art couvre l'utilisation de la domotique à travers les projets de recherche et d'habitats intelligents [Chan *et al.* 2008, Chan *et al.* 2009, Noury *et al.* 2003]. Ces derniers servent de support à des équipes de recherche pour implémenter et/ou tester en situation réelle leurs méthodes et outils développés. S'il ne prétend pas être exhaustif, cet état de l'art permet tout de même de lister un certain nombre de projets qui ont eu un impact dans la recherche en domotique ces dernières années et qui feront les habitats de demain. Un exemple de méthodes et/ou d'outils développés est donné pour chacun de ces projets.

Au Colorado (Boulder), un habitat intelligent appelé *Neural Network House* a été développé pour permettre le contrôle adaptatif de la température, du chauffage et de la luminosité sans programmation préalable par les occupants [Mozer 1998]. Basé sur les

réseaux de neurones, le système ACHE (*Adaptive Control of Home Environment*) réalise un apprentissage des activités des occupants à partir des informations de capteurs distribués dans l'habitat. Ce système est chargé de répondre simultanément à deux objectifs : économiser de l'énergie dans l'habitat par une gestion efficace des équipements électriques tout en respectant les habitudes de vie et les souhaits de l'utilisateur. Le premier objectif est de *satisfaire* et d'*anticiper* les besoins de l'utilisateur pour l'éclairage, la température de l'air et la ventilation. Un ajustement manuel du système est interprété par une insatisfaction de la configuration courante et sert de données pour l'apprentissage des habitudes de l'utilisateur. Le deuxième objectif est de *réduire* la consommation d'énergie de l'habitat. Les lumières doivent être utilisées avec le minimum d'intensité requise, l'arrivée d'eau chaude doit être maintenue à une température minimum pour satisfaire la demande, etc.

Le projet *MavHome (Managing an Adaptive Versatile Home)* de l'université du Texas (Arlington) [Das *et al.* 2002, Cook *et al.* 2003] définit la maison comme un agent rationnel chargé de maximiser le confort de ces habitants tout en minimisant le coût des opérations de recherche. L'architecture générale du système se base sur une hiérarchie d'agents logiciels qui coopèrent pour atteindre leurs objectifs. Ils sont plus précisément chargés de prédire la localisation de l'utilisateur et ses interactions avec le système pour l'automatisation des décisions, le routage des informations pour l'affichage de messages à l'utilisateur, etc. Pour la localisation de l'utilisateur, un modèle probabiliste de prédiction de chemins est défini. L'habitat est ainsi vu comme un graphe pondéré et orienté où les noeuds représentent des zones et les arcs une jonction (murs, couloirs, etc.) entre deux zones. L'algorithme *Active LeZi* [Bhattacharya & Das 2002] calcule ainsi la probabilité d'apparition de chaque mouvement de l'utilisateur qui peut survenir dans la séquence en cours d'observation. Il prédit ensuite le mouvement qui a le taux de probabilité le plus important pour exécuter les services adéquats.

En Floride, un habitat intelligent connu sous le nom de *GatorTech Smart House* est équipé de nombreux équipements *augmentés* ("*smart devices*") pour assister des personnes âgées dépendantes [Helal *et al.* 2005]. Par exemple, le miroir de la salle de bain peut afficher des messages importants ou des rappels pour la prise de médicaments si besoin. La porte d'entrée intègre quant à elle une identification par radio-fréquence (tag *RFID*) pour un déverrouillage et une ouverture sans clé par les résidents ou le personnel autorisé. Le plancher de la cuisine est équipé de capteurs de pression pour identifier et suivre la localisation des occupants mais aussi détecter les chutes. L'architecture logicielle orientée service, basée sur les *bundles OSGI*, offre la possibilité d'intégrer dynamiquement de nouveaux équipements.

*Aware Home* [Abowd *et al.* 2002] du *Georgia Institute of Technology* est un laboratoire d'expérimentation de 1536m<sup>2</sup> composé de trois étages. Il sert de plateforme pour la conception, le développement et l'évaluation de technologies domotiques innovantes. Les chercheurs de ce projet ont développé entre autres un système de suivi et de détection pour aider à retrouver des objets qui se perdent fréquemment comme des porte-feuilles,

des lunettes et des télécommandes. Chacun de ces objets se voit attribuer un tag *RFID* permettant de les identifier. L'utilisateur interagit avec son système, qui le guide pour retrouver des objets perdus en utilisant des indications sonores.

"*The House of the future*" (*House\_n*) du MIT [Intille 2002] propose un système de propositions de services en prenant en compte les facteurs environnements et le comportement des utilisateurs. Le système comprend trois modules : i) un ensemble de capteurs utilisés pour collecter des informations sur l'utilisation des équipements domotiques, ii) des techniques d'*ESM* (*Experience Sampling Method*)<sup>7</sup> [Intille *et al.* 2002] utilisées par le sujet pour labeliser ses activités et iii) un module de reconnaissance d'activités. Les utilisateurs sont équipés d'un assistant électronique personnel de type *PDA* (*Personal Digital Assistant*) sur lequel tourne l'application *ESM* qui leur demande de renseigner leurs activités [Intille *et al.* 2002]. La figure 1.7 montre une application de monitoring de l'utilisateur qui se base sur l'enregistrement vidéo à partir de caméras. Ces dernières capturent des images de l'utilisateur sans le perturber dans la réalisation de ses activités de la vie quotidienne. Plus tard, ces images lui sont montrées sur son *PDA* pour qu'il puisse les annoter en décrivant l'activité qu'il était en train de réaliser ou en décrivant ses impressions à cet instant. Ces données sont par la suite utilisées par les chercheurs pour évaluer leurs algorithmes de reconnaissance d'activités ou pour adapter automatiquement des services à l'utilisateur.

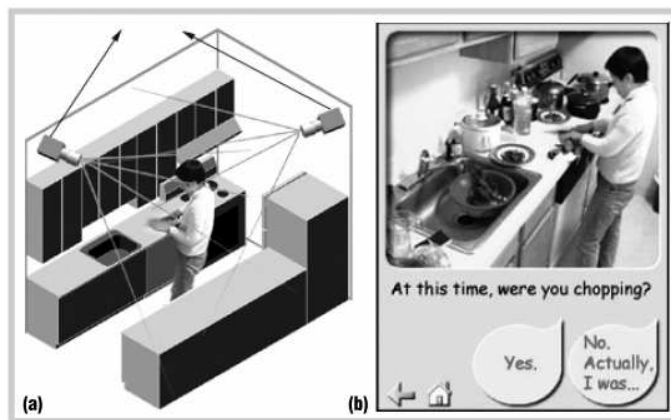


FIGURE 1.7 – Installation du système de tracking vidéo (a) et interface utilisateur (b) de l'application "*Image-based experience sampling*" du MIT

*Elite Care* (*Creating an Autonomy-Risk Equilibrium*) est une résidence localisée à Portland, entièrement équipée en domotique [Elite Care]. Cette résidence est habitée par des personnes retraitées qui souffrent pour la majorité d'entre elles de démence de type

7. *Experience Sampling Method (ESM)* : techniques de recueil d'informations basées sur les annotations utilisateur pour enregistrer en temps réel leurs comportements, pensées ou impressions.

Alzheimer. L'objectif ici est de prolonger l'indépendance de la personne via la domotique, mais aussi de permettre aux aides soignants et à l'équipe paramédicale d'identifier les problèmes de santé le plus tôt possible. Les chercheurs associés au projet ont développé une méthode de monitoring non-intrusif des mouvements lors du sommeil. Les données recueillies par des cellules de charge (*load cells* utilisées pour convertir une force en signal électrique) permettent d'estimer l'heure du coucher, l'heure de réveil et les changements de position du résident pendant son sommeil [Adami *et al.* 2003].

À Toronto, au Canada, un système de tracking vidéo capable de suivre les gestes de personnes âgées dépendantes a été développé [Mihailidis *et al.* 2004]. Le système de vision est composé de trois agents logiciels : un agent chargé de la détection, un agent pour la planification et le dernier pour la proposition de services. Des méthodes de reconnaissance de la segmentation de la couleur de la peau (pour éviter l'utilisation de marqueurs) sur des images numériques ont été utilisées pour reconnaître le visage et la position des mains en temps réel. L'objectif principal de ce projet n'est pas seulement de suivre les positions des mains associées à chaque activité de la vie quotidienne (AVQ), mais aussi de favoriser, de la "*manière la moins intrusive possible*", le maintien à domicile des personnes âgées dépendantes.

Le laboratoire Domus de l'université de Sherbrooke au Québec étudie l'assistance cognitive, le suivi médical et la télévigilance auprès des personnes avec troubles cognitifs. Les résultats de ces recherches sont applicables aux personnes atteintes de déficits cognitifs imputables en particulier à des traumatismes crâniens, à la schizophrénie, à la maladie d'Alzheimer et à la déficience intellectuelle [Kadouche *et al.* 2010, Domus]. Dans [Chikhaoui *et al.* 2010], les auteurs proposent une approche de détection de personnes basée sur l'extraction de motifs séquentiels et récurrents à partir de données de capteurs enfouies dans l'environnement. Sans utiliser de captures audio-visuelles, cette approche se décompose en trois parties : i) l'extraction de motifs séquentiels, ii) l'attribution de poids aux motifs extraits et iii) la classification. Des expérimentations ont été réalisées à partir des jeux de données [Domus] collectés dans le laboratoire.

L'habitat intelligent porté par le professeur Matsuoka, localisée à Osaka au Japon [Zhang & MoUuari 2004], détecte automatiquement des événements inhabituels qui peuvent faire référence à des accidents domestiques (e.g. une chute). Cette détection tire bénéfice d'un réseau de 167 capteurs installés dans l'habitat. On y retrouve des tags *RFID*, des capteurs électro-magnétiques sur toutes les portes ou fenêtres et des caméras et détecteurs de mouvements infrarouges dans toutes les pièces. Des capteurs portés par la personne sont aussi utilisés : un accéléromètre permet par exemple de détecter les postures du résident (debout, assis, couché, en train de marcher, en train de chuter). Chaque capteur est relié à une ou plusieurs activités : se lever, aller au lit, préparer un repas, prendre une douche, travailler au bureau, etc. Les situations inhabituelles sont alors détectées lorsque i) une activité dure trop longtemps (fixé par un seuil), ii) le rythme des activités se dégrade et iii) des comportements inhabituels surviennent.

Le projet GerHome [GerHome] a pour objectif de concevoir, d'expérimenter et de certifier des solutions techniques favorisant l'aide au maintien à domicile de personnes âgées pour assurer autonomie, confort de vie, sécurité, surveillance et assistance domicile. Par exemple, dans [Zouba *et al.* 2009], un système de monitoring basé sur des caméras et différents types de capteurs permet d'analyser le comportement des habitants et leurs déplacements. La fusion d'information multi-sources permet d'aboutir à une précision pour la reconnaissance d'activités qui varie de 62 à 94% sur un jeu de données produits par deux volontaires (64 et 85 ans) dans l'appartement GerHome. Comme une extension, le projet GerHomeLabs propose une mise en situation réelle (in vivo) pour l'expérimentation des dispositifs développés dans le cadre du projet Gerhome, en les déployant dans deux établissements hébergeant des personnes âgées dépendantes (EHPAD de Nice et Antibes) et dans deux logements privés favorisant le maintien et l'aide à domicile (Nice et Valbonne).

Le projet SIGAAL (Services Inter Générationnels pour l'Assistance aux Aînés dans leur Logement) porte également sur l'aide au maintien à domicile de personnes âgées dépendantes [SIGAAL]. Ce projet vise à réduire l'isolement de ces personnes en leur offrant des services personnalisés pour maintenir ou pour renforcer les liens sociaux. Dans le cadre de ce projet, [Lézoray *et al.* 2011] propose un processus de conception permettant l'adaptation dynamique de services dans un contexte d'informatique diffuse. En utilisant les méthodes de développement basées sur l'ingénierie dirigée par les modèles, l'approche de "medium" est proposée. Il s'agit d'interconnexions de composants logiciels qui peuvent être de différentes natures suivant le service considéré mais aussi la nature des composants qui y sont connectés.

### 1.2.2.2 Projets du Lab-STICC / MOCS

L'équipe MOCS (*Méthodes, Outils pour Circuits et Systèmes*) du laboratoire Lab-STICC participe depuis plusieurs années à des projets portant sur la domotique pour l'assistance aux personnes dépendantes. L'axe de recherche "systèmes pervasifs pour l'aide au handicap" regroupe des travaux à finalité applicative, des flots méthodologiques pour le déploiement de systèmes domotiques qui doivent être le moins intrusif possible tout en garantissant un niveau de qualité de services acceptable. Cette thématique s'appuie sur une collaboration étroite de plusieurs années avec le Centre de Rééducation et de Réadaptation Fonctionnelles (CMRRF) de Kerpape qui dispose de nombreuses infrastructures domotiques. La Région Bretagne et le Conseil Général du Morbihan portent un intérêt particulier à la problématique de la dépendance et financent depuis quelques années des projets qui visent à y apporter des solutions technologiques.

Sept projets ont vu le jour depuis la création de cet axe thématique. En 2003, les premières briques sont posées avec la thèse d'Aïssam Belabbas [Belabbas 2007] (projet COHAND) aboutissant ainsi à une plateforme de simulation pour la navigation de fauteuils et la proposition conjointe de services domotiques en présence d'aléas. Deux



ans plus tard, un projet est monté dans le but de définir une architecture matérielle à déployer à échelle réelle au centre de Kerpape. Le projet *Quatra* regroupe ainsi différents acteurs du domaine tels que le Lab-STICC, le Valoria et le CMRRF de Kerpape [De Lamotte *et al.* 2008]. En 2006, la thèse de Saïd Lankri [Lankri 2009] propose de développer les approches proposées dans les deux précédents projets. Le middleware *DANAH* est développé pour permettre le contrôle à distance et l'automatisation des différents équipements domotiques de manière indépendante du protocole de communication utilisé. En parallèle de ces travaux, la thèse de Thanh Truong (*AIDOM*) [Truong 2010] adopte une approche originale pour la détection automatique de scénarios de vie à partir des commandes émises par l'utilisateur. L'algorithme développé peut être gourmand en ressources (occupation mémoire, processeur, etc.) et de ce fait peut altérer la qualité de services du système domotique. Pour répondre à cette problématique, le projet *STAD*, qui débute en 2010, tente d'apporter des solutions d'ordonnancement et de déploiement dynamique de services en fonction des capacités physiques des systèmes électroniques présents dans l'environnement. Cette thèse (*INTELHOME*) fait suite aux travaux de thèse *DANAH* et *AIDOM* et a pu bénéficier d'un transfert technologique dans le cadre du projet *ASIM*.

**COHAND** Ces travaux ont porté sur le contrôle environnemental et la mobilité des personnes en situation de handicap en étudiant le couple "Offre de services / Navigation du fauteuil électrique" [Belabbas *et al.* 2006a, Belabbas *et al.* 2006b]. Lors de la sélection d'un service, le système pilote le fauteuil et les services associés à la navigation (e.g. ouvrir les portes, allumer les lumières) pour finalement activer le service sollicité (e.g. regarder la télévision). La reconfiguration en cas de panne joue un rôle important pour garantir une certaine qualité de services ( $QoS$ <sup>8</sup>). Basée sur l'exploitation de modèles topologiques représentant l'environnement, l'approche proposée permet de calculer le chemin approprié pour accéder au service souhaité. En présence d'aléas (obstacles ou mauvaises exécutions de services), le système est capable de se reconfigurer (changement de chemin et utilisation d'équipements aux capacités similaires) pour fournir malgré tout le service demandé. Ces travaux ont pu être évalués sur une plate-forme de simulation.

**Quatra** Ce projet aboutit à une véritable plate-forme expérimentale testée au CMRRF de Kerpape [De Lamotte *et al.* 2008]. Une architecture de bornes domotiques communicantes a été définie pour assurer la délivrance de services domotiques. L'utilisateur, équipé de son *PDA* (*Personal Digital Assistant*) peut se connecter i) à une borne mobile embarquée dans son fauteuil pour commander ce dernier et ii) aux bornes fixes distribuées dans l'environnement domotique. Le démonstrateur réalisé a permis de tester un ensemble de scénarios répondant aux besoins des patients du centre. Dans le but de faciliter l'accès aux services de la vie quotidienne, l'utilisateur peut contrôler ses équipements domotiques (volets, portes) à travers une interface unique où un ensem-

---

8.  $QoS$  : vient du terme anglais *Quality of Service*

ble de services élémentaires et scénarios lui sont publiés. Lors de l'activation d'un scénario, le système génère automatiquement la séquence de commandes à activer pour satisfaire la demande de l'utilisateur.

**DANAH** DANAH (Domotique, Aide à la Navigation et Assistance au Handicap) reprend les différentes problématiques scientifiques issues des projets précédents pour améliorer les propositions qui ont été faites. Le middleware du même nom permet de faire abstraction des couches basses de communication pour se concentrer sur la notion de services proposés à la personne. Il intègre un processus de reconfiguration multi-niveaux [Lankri *et al.* 2009] : reconfiguration statique / dynamique de services domotiques et reconfiguration de chemins pour le déplacement en fauteuil roulant électrique. Un premier flot de conception, basé sur une approche modulaire a de plus été introduit. La modélisation de l'environnement domotique par l'expert en charge de la configuration logicielle peut être transformée pour aboutir à un code fonctionnel implantable sur des bornes domotiques intégrant le middleware DANAH. La distribution de l'architecture matérielle est mise en avant et apporte une contribution supplémentaire [Lankri *et al.* 2008]. Elle permet, entre autres, de faire face aux problèmes de distance liés aux limites de portée de certaines technologies sans fils.

**AIDOM** En parallèle des travaux menés autour des systèmes de contrôle domotique, cette thèse s'intéresse à la fouille de données pour la détection automatique de scénarios de vie usuels [Truong *et al.* 2009a, Truong *et al.* 2009b]. L'originalité des travaux réside dans l'analyse des activités domotiques de l'habitant sans équipements supplémentaires (e.g. capteurs, caméras), mais en utilisant les ordres de commande (services sollicités) que l'utilisateur transmet au système domotique. La qualité de services perçue par l'utilisateur, qui peut être évaluée par un ergothérapeute, est mise en relation avec l'énergie déployée par la personne dépendante lors de l'activation de services. Le critère d'évaluation retenu est donc l'énergie déployée par l'utilisateur que l'algorithme cherche à minimiser en proposant des scénarios pour l'activation automatique de services. Dans ces travaux, chaque service est vu comme un motif qui se répète dans le temps. Une analyse fréquentielle des services sollicités permet ainsi de proposer des scénarios mais aussi de lancer une alerte si jamais la personne dévie trop de ses activités de vie quotidienne [Truong *et al.* 2010].

**IntelHome** Le travail présenté dans ce mémoire fait suite aux projets DANAH et AIDOM, et vise à définir une méthodologie de conception pour les systèmes domotiques d'assistance. Les problématiques associées à la conception de ces systèmes, auxquelles tente de répondre ce projet de thèse, sont présentées dans la section suivante. Les contributions scientifiques sont quant à elles précisées dans la section 1.3.2.

**ASIM** (*Assistance par Serveur Intelligent Multiprotocoles*) est un projet “santé et autonomie sur le lieu de vie grâce au numérique” des investissements d’avenir. Porté par l’entreprise *Vity Technology* et en collaboration avec l’entreprise *Kaptalia* et l’*Université de Bretagne Sud*, ce projet concerne la réalisation d’un serveur domotique multi-protocoles capable d’interpréter des données issues de capteurs bio-médicaux avec l’objectif de faciliter le maintien à domicile des personnes vulnérables. Le logiciel de configuration bénéficie d’un transfert technologique des travaux présentés dans ce mémoire.

L’équipe MOCS est de plus impliquée dans le projet transversal et pluri-disciplinaire *HAAL (Human Ambient Assisted Living)* qui regroupe différentes équipes du Lab-STICC s’intéressant à la thématique de l’aide à la personne.

### 1.2.3 Verrous scientifiques identifiés

L’état de l’art des projets domotiques pour l’assistance aux personnes dépendantes de la section précédente témoigne de la dynamique autour de ce domaine. Ces projets mettent en évidence un certain nombre de verrous, qu’ils soient spécifiques aux systèmes domotiques d’assistance (1 à 5) ou bien qu’ils interviennent dans l’acceptation générale d’une solution domotique (6 à 8) :

1. **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.
2. **Scalabilité** : 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).
3. **Disponibilité** : en lien étroit avec la sécurité, la sûreté de fonctionnement et en particulier la disponibilité en cas de panne est une caractéristique essentielle qui doit être prise en compte pour assurer la délivrance d’un service.
4. **Interopérabilité** : il faut faire face à l’hétérogénéité des technologies et protocoles de communication présents dans un habitat, ainsi que ceux qui seront à intégrer.
5. **Évolutivité** : de nouveaux protocoles de “haut niveau” doivent pouvoir être intégrés sans avoir à modifier le travail de conception initial.
6. **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.
7. **Acceptabilité/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 doit en aucun cas perturber l’intimité d’un individu, ni même stigmatiser l’utilisateur (porteur de handicap).
8. **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.

### 1.3 Motivations & Contributions

Les motivations de ces travaux ainsi que les contributions scientifiques sont précisées dans cette section. Nous présentons dans un premier temps les motivations principales qui ont permis de donner une orientation aux recherches présentées dans ce mémoire (section 1.3.1). En réponse à ces motivations (réflexions suite aux intuitions de départ) et en réponse aux verrous de la section précédente, nous présentons par la suite les contributions scientifiques apportées (section 1.3.2).

#### 1.3.1 Motivations

##### 1.3.1.1 Conception universelle / centrée utilisateur

La conception universelle (également appelée *Design For All* ou *Design Universel*) pose comme principe qu'un produit idéalement conçu est susceptible d'être utilisé par tous, quels que soient ses capacités, son âge ou sa culture. Cette philosophie de conception part du constat que le design grand public se préoccupe peu de l'utilisateur [Pruski 2003] (c'est à ce dernier de s'adapter au système). La conception universelle a pour but de renverser cette tendance en proposant au système (matériel ou logiciel) d'être en adéquation avec les aptitudes physiques et intellectuelles du plus grand nombre d'utilisateurs possible. L'application systématique de ces principes donne une plus grande chance aux personnes en situation de handicap d'accéder aux mêmes fonctionnalités que les autres usagers.

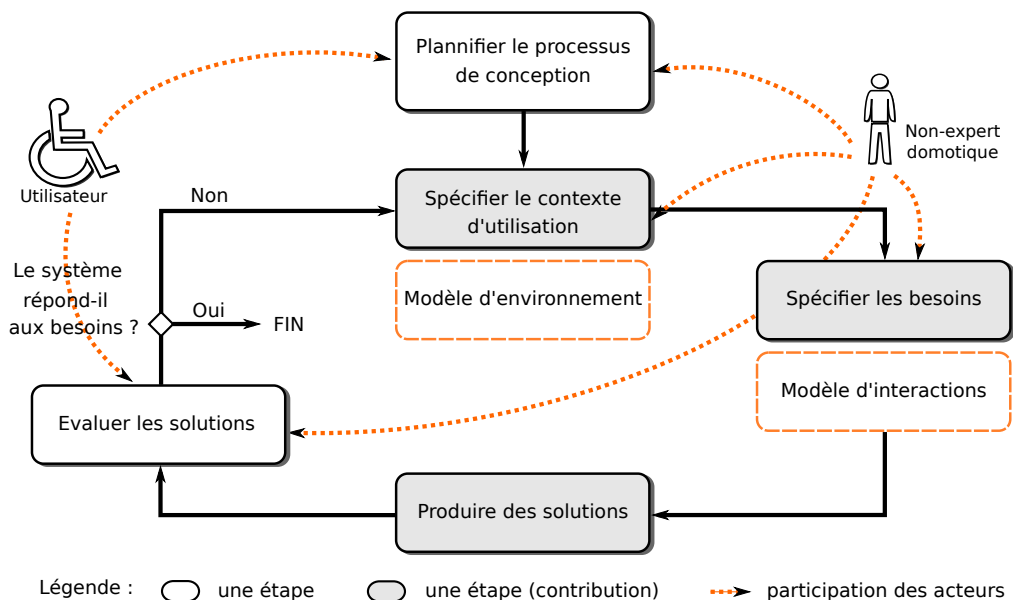


FIGURE 1.8 – Processus de conception centré utilisateur pour la domotique

L'application méthodologique de cette philosophie s'oriente naturellement vers un processus de conception centré utilisateur [Maguire 2001] (*Human Centered Design*, ISO 13407). Ce processus ergonomique permet de prendre en compte les besoins de l'utilisateur à chaque étape de la conception d'un nouveau produit. Son développement est ainsi d'avantage guidé par les besoins et les exigences des utilisateurs finaux que par des possibilités techniques et technologiques. Ce processus itératif est basé sur quatre étapes fondamentales (cf figure 1.8, sauf parties oranges) : i) la phase d'analyse qui vise à préciser les attentes et les besoins des utilisateurs finaux, ii) la phase de conception séparée en spécifications du contexte d'utilisation et des besoins associés, iii) la phase de production des solutions conformément à ce qui a été défini dans les étapes précédentes et iv) la phase d'évaluation pour mesurer l'utilisabilité du produit.

En adoptant la philosophie du *Design For All* pour la domotique d'assistance, appliquée à un processus de conception centré utilisateur, il est possible d'apporter des réponses essentiellement à la problématique 1 de la section 1.2.3, à savoir l'**adaptabilité** : l'utilisateur doit bénéficier de services adaptés à ses besoins, intimement liés à sa situation de handicap. Les situations de handicap étant variées, l'application domotique doit idéalement pouvoir s'adapter à chacune d'entre elles.

#### 1.3.1.2 Nouveau schéma de collaboration

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. Pour l'assistance aux personnes dépendantes, une tierce personne (e.g. ergothérapeute, membre de la famille) est généralement 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 1.9). 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 toute autre personne intimement liée à l'utilisateur (dans la confiance de ses besoins, e.g. membre de la famille), plutôt que par un domoticien.

En modifiant le schéma de collaboration pour la conception d'un système domotique d'assistance comme présenté sur la figure 1.9, il est possible de prendre en compte

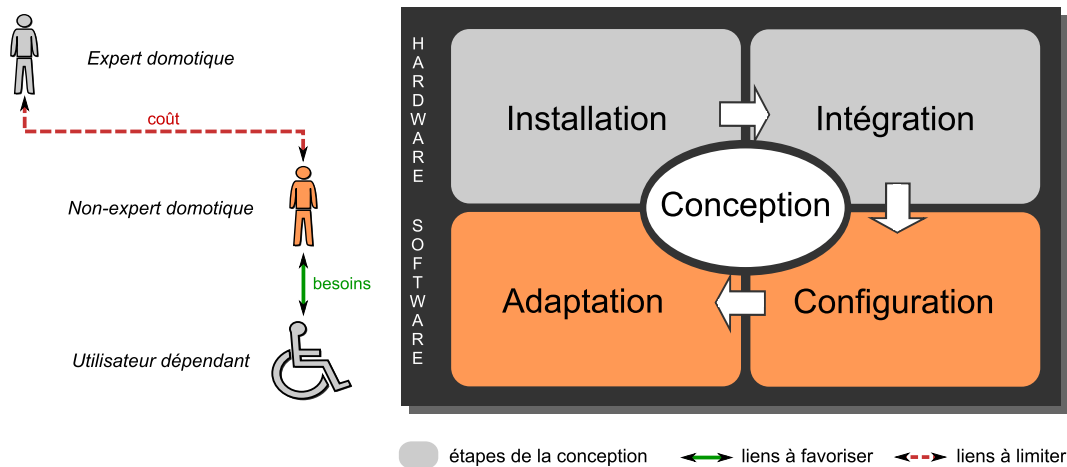


FIGURE 1.9 – Schéma de collaboration pour la conception domotique d’assistance

essentiellement les problématiques 6 et 8 de la section 1.2.3 :

- **Prix** : l’intervention systématique d’un expert à chaque fois que les besoins de l’utilisateur évoluent peut être substituée par l’intervention de ce concepteur non-expert. Les coûts associés à la configuration logicielle peuvent ainsi être réduits (pour la configuration d’un nouvel équipement, la création de services domotiques, etc.).
- **Facilité de déploiement** : la configuration logicielle, actuellement réservée à des domoticiens, doit donc être accessible à des personnes non-expertes en informatique ou électronique pour une meilleure intégration des besoins de l’utilisateur.

### 1.3.1.3 Supervision non-intrusive d’interactions

La course à la miniaturisation des équipements électroniques et leur diffusion dans le milieu ambiant fait que la domotique ne se cantonne plus au simple contrôle d’équipements domestiques. Ce domaine d’application évolue de plus en plus vers les systèmes pervasifs [Weiser 1993]. Dans ce contexte, savoir *qui fait quoi* et *où* est essentiel pour adapter un comportement “intelligent”, spécialement dans le domaine de l’assistance à la personne [Chan *et al.* 2008, Chan *et al.* 2009, Noury *et al.* 2003]. Pour permettre d’adapter le fonctionnement du système en fonction des besoins de l’utilisateur, et ce de manière dynamique, il faut pouvoir recueillir et traiter un certain nombre d’informations contextuelles. Ces informations peuvent être de différentes natures : localisation, préférences, repères temporels, informations de santé dans le cadre de la télémédecine ou d’un suivi bio-médical, etc.

L’utilisation de capteurs distribués dans l’environnement est actuellement la solution la plus répandue pour recueillir ces informations. On peut distinguer principalement deux méthodes [Nourizadeh *et al.* 2009, Yu *et al.* 2006], qui peuvent être cou-

plées. Si ces méthodes sont efficaces pour faire du monitoring d'activités, elles posent quand même un certain nombre de problèmes quand on s'intéresse à l'acceptabilité de la solution domotique par l'utilisateur final. La première méthode est basée sur l'utilisation de caméras qui représente une solution intrusive, et de ce fait, a beaucoup de difficultés à être acceptée par les utilisateurs souhaitant naturellement garder un certain degré d'intimité. La deuxième méthode consiste à utiliser massivement des capteurs (electro-magnétiques, infrarouges, ultrasons, etc.) distribués dans l'environnement. Cette dernière approche est limitée par un critère majeur qui intervient dans l'acceptation d'une solution domotique, à savoir le coût de ces capteurs spécifiques ainsi que celui de l'installation et de la maintenance de ces équipements. Quoiqu'il en soit, nous pensons qu'une solution domotique limitant les modifications physiques (i.e. visuelles) d'un habitat doit être privilégiée pour une meilleure acceptabilité, notamment auprès de personnes âgées dépendantes.

Nous pensons que l'analyse des commandes envoyées par l'utilisateur à partir de son interface de contrôle domotique peut donner des informations suffisamment pertinentes pour limiter l'utilisation de capteurs dans le cas où on cherche à réduire leurs coûts. Dans le cas où on cherche à augmenter la sûreté de fonctionnement, l'analyse des commandes peut aussi s'avérer utile pour générer des informations redondantes par rapport aux informations issues de capteurs. Pour résumer, une approche non-intrusive basée sur l'analyse des interactions entre l'utilisateur et son système domotique peut apporter des réponses aux problématiques 3, 6 et 7 de la section 1.2.3 :

- **Disponibilité** : l'analyse des commandes émises par l'utilisateur, en parallèle de l'analyse des capteurs permet d'avoir de la redondance d'informations qu'il est possible par la suite d'exploiter pour garantir la sûreté de fonctionnement (pour détecter la présence de pannes et amorcer un processus de reconfiguration).
- **Prix** : l'analyse de certaines commandes domotiques permet de se substituer au capteurs déployés massivement. Cela permet de limiter le coût relatif à l'installation et la maintenance de ces équipements.
- **Acceptabilité/Intrusivité** : l'approche de supervision non-intrusive prône une utilisation de capteurs la plus faible possible et de ce fait permet de diminuer l'intrusivité de la solution domotique.

## 1.3.2 Contributions scientifiques

### 1.3.2.1 Langage de modélisation domotique

Le langage de modélisation proposé doit répondre aux besoins des deux acteurs identifiés dans nos travaux. Il s'agit du *concepteur* et de l'*utilisateur*. Dans le cadre de l'assistance à la personne, ces deux acteurs présentent des profils originaux par rapport au contexte standard d'utilisation de la domotique.

**Point de vue utilisateur** L'utilisateur peut présenter des capacités motrices réduites ou des troubles cognitifs (voire une combinaison des deux), à des stades plus ou moins avancés. À la pluralité des situations de handicap (cf section 1.1.2.2), le langage de conception domotique proposé dans ces travaux doit être suffisamment expressif pour répondre à un nombre de besoins le plus large possible. En outre, des interactions de haut niveau sémantique sont essentielles pour faciliter et/ou automatiser le fonctionnement de l'habitat et compenser ainsi une situation de handicap.

**Point de vue concepteur** 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. Pour cette personne, un système domotique d'assistance doit avant tout être facile à concevoir, mais aussi facile à maintenir, facile à commander et flexible. Elle a donc besoin d'outils de conception/de modélisation adaptés à ses compétences.

Ces besoins originaux peuvent être résumés ainsi : il faut élever le niveau d'abstraction, tant au niveau de la conception que de l'utilisation du système domotique. Pour cela, nous proposons un langage de modélisation de haut niveau pour la description de ces systèmes (*DSML : Domain Specific Modeling Language*), composé de concepts abstraits de toutes spécificités matérielles (abordables pour des personnes non-experte en domotique), et avec un pouvoir d'expression permettant de faire face à la pluralité des situations de handicap. Ce langage est composé d'une vue physique permettant de représenter l'environnement domotique et d'une vue fonctionnelle pour la prise en compte des besoins de l'utilisateur (interactions utilisateur-système).

### 1.3.2.2 Flot de conception dirigé par les modèles

À partir des modèles définis avec le langage de modélisation proposé dans ces travaux, nous proposons de générer automatiquement le code de commande de systèmes domotiques. Pour cela, nous définissons une méthodologie de conception intuitive (cf figure 1.8, partie orange) basée sur un processus de conception centré utilisateur. 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.



Afin de permettre la mise en oeuvre informatique d'un tel processus, nous utilisons l'IDM. Ce paradigme pose comme principe que tout ou partie d'une application est généré à partir de modèles. Le code source d'une application logicielle n'est donc plus considéré comme l'élément central d'un logiciel, mais comme un élément dérivé d'éléments de modélisation. Cette élévation d'abstraction permet ainsi de réduire la complexité des systèmes informatiques tant au niveau de la conception que de la validation. L'utilisation du DSML proposé permet au concepteur de manipuler des modèles indépendants de toute plateforme, pour 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 (code de commande) conformément aux spécifications préalablement définies.

### 1.3.2.3 Modèle de supervision basé sur les ontologies

Des travaux de thèse du laboratoire Lab-STICC ont pu mettre en avant l'intérêt de réaliser du monitoring en se basant sur les logs de commande (commandes infrarouges émises depuis la téléthèse de l'utilisateur). Dans son modèle de supervision, [Truong *et al.* 2009b] interprète un service domotique sollicité par l'utilisateur comme un motif qui se répète dans le temps. En pratique [De Lamotte *et al.* 2008, Truong 2010], on s'aperçoit qu'un certain nombre de services domotiques ne respectent pas cette loi temporelle. Par exemple, l'utilisateur a besoin d'allumer ou d'éteindre les lumières dans les pièces qu'il traverse pour aller aux toilettes la nuit. Ce genre de services peut être sollicité de manière ponctuelle et non-régulière : à n'importe quelle heure de la nuit, à n'importe quelle journée de la semaine, etc.

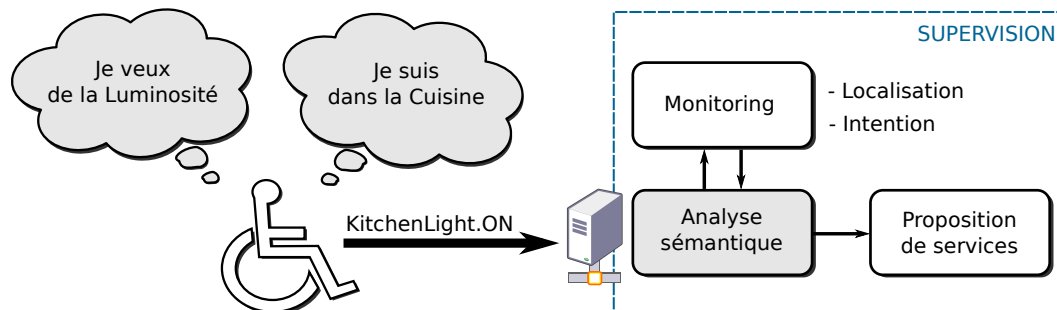


FIGURE 1.10 – Principe général pour l'analyse sémantique des logs domotiques

Une analyse sémantique des commandes permet alors de prendre en compte ce genre de services mais aussi de générer des informations pertinentes pour le monitoring de l'utilisateur. De manière instinctive, on peut déduire que lorsque l'utilisateur

décide d'activer une lumière dans une pièce (cf figure 1.10), c'est i) qu'il veut de la lumière et ii) qu'il se trouve dans cette pièce. L'analyse sémantique de la commande "*KitchenLight.ON*" doit permettre de retrouver ce genre d'informations. Cette analyse peut ainsi servir pour le monitoring des activités de la personne comme pour la proposition de services : on peut par exemple inférer qu'à la première activation d'une lumière dans une pièce, l'utilisateur en aura besoin dans la pièce voisine dans laquelle il s'apprête à entrer. Pour modéliser et interpréter la sémantique associée aux commandes domotiques, nous proposons d'utiliser une ontologie. Ce modèle de représentation de connaissances nous permet de définir les interactions entre l'utilisateur et son système domotique. Un moteur d'inférences interprète par la suite les liens sémantiques de cette ontologie pour déduire la localisation de l'utilisateur et son intention (par raisonnement logique) pour aboutir à la proposition de services.

Les contributions de cette thèse, présentées sommairement dans cette section, font chacune l'objet d'un chapitre du mémoire. Elles sont ensuite validées dans le chapitre 5 : une série d'expérimentations est conduite pour valider le DSML (chapitre 2) et le flot de conception (chapitre 3) sur le cas d'étude des appartements tremplins du centre de Kerpape. Le modèle de supervision non-intrusif (chapitre 4) est quant à lui validé sur les jeux de données tirés de l'habitat intelligent Domus de l'Université de Sherbrooke.



# 2

## DSML pour la domotique d'assistance

*"If modeling makes things more complex,  
you're doing something wrong"*

Ed Merks, Eclipse EMF project leader

L'Ingénierie Dirigée par les Modèles (IDM) vise à réduire la complexité des systèmes informatiques via l'utilisation de représentations simplifiées, de techniques et d'outils permettant la génération automatique de tout ou partie de ces systèmes. Les langages de modélisation, qu'ils soient généralistes ou dédiés, jouent un rôle fondamental dans ce processus de développement. Dans ces travaux, nous proposons un langage de modélisation (*DSML : Domain Specific Modeling Language*) de haut niveau d'abstraction pour qu'une personne non-experte en domotique puisse intervenir dans les étapes de la conception domotique. Ce langage se veut le plus expressif possible pour répondre à la diversité des besoins que peuvent exprimer les personnes en situation de handicap. L'objectif de ce chapitre est de présenter le langage de modélisation proposé dans ces travaux. La section 2.1 introduit l'IDM et les langages de modélisation. Un état de l'art sur les approches de modélisation pour la conception de systèmes domotiques est ensuite présenté dans la section 2.2, permettant de positionner cette contribution par rapport à l'existant. Ce langage de modélisation fait l'objet de la section 2.3, où les concepts le constituant y sont détaillés.

## Sommaire

---

|         |  |    |
|---------|--|----|
| 2.1     | Introduction . . . . .                                   | 45 |
| 2.1.1   | L'Ingénierie Dirigée par les Modèles . . . . .           | 45 |
| 2.1.1.1 | Organisation hiérarchique des modèles . . . . .          | 45 |
| 2.1.1.2 | Transformation de modèles . . . . .                      | 48 |
| 2.1.2   | Les langages de modélisation . . . . .                   | 49 |
| 2.1.2.1 | Définition . . . . .                                     | 49 |
| 2.1.2.2 | Les langages dédiés . . . . .                            | 50 |
| 2.1.2.3 | Langage et MDA . . . . .                                 | 51 |
| 2.2     | État de l'art & positionnement . . . . .                 | 53 |
| 2.2.1   | Modélisation spécifique à une plateforme . . . . .       | 53 |
| 2.2.2   | Modélisation généraliste indépendante . . . . .          | 53 |
| 2.2.3   | Modélisation de domaine indépendante . . . . .           | 54 |
| 2.2.4   | Choix retenu . . . . .                                   | 55 |
| 2.3     | Description du DSML . . . . .                            | 56 |
| 2.3.1   | Méta-modèle de l'environnement . . . . .                 | 56 |
| 2.3.1.1 | Modélisation de l'environnement par composants . . . . . | 56 |
| 2.3.1.2 | Une approche ascendante de conception . . . . .          | 61 |
| 2.3.2   | Méta-modèle des interactions . . . . .                   | 64 |
| 2.3.2.1 | Les services . . . . .                                   | 64 |
| 2.3.2.2 | Les modes . . . . .                                      | 67 |
| 2.3.2.3 | Activation et définition . . . . .                       | 71 |
| 2.4     | Conclusion du chapitre / contribution . . . . .          | 72 |

---

## 2.1 Introduction

L'Ingénierie Dirigée par les Modèles (IDM), ou *Model-Driven Engineering (MDE)* pose comme principe que tout ou partie d'une application informatique est généré à partir de modèles. Cette approche est fondée sur l'utilisation systématique de modèles tout au long du processus de développement logiciel (section 2.1.1). Tout modèle se définit à l'aide d'un langage de modélisation (section 2.1.2) qui doit lui-même être rigoureusement défini pour répondre aux besoins précis d'un domaine.

### 2.1.1 L'Ingénierie Dirigée par les Modèles

L'IDM fournit un cadre de développement logiciel dans lequel les modèles passent de l'état passif (contemplatif) à l'état actif (productif) et deviennent les éléments de base dans le processus de développement de logiciels. Ce processus peut ainsi être vu comme un ensemble de transformations de modèles, où chaque transformation prend des modèles en entrée et produit des modèles en sortie, jusqu'à l'obtention d'artéfacts exécutables. Cette démarche, basée sur une organisation hiérarchique de modèles et sur les transformations s'y appliquant, met ainsi le modèle au centre des préoccupations des analystes/concepteurs.

#### 2.1.1.1 Organisation hiérarchique des modèles

L'IDM repose sur une hiérarchisation des modèles manipulés qui fait intervenir deux concepts fondamentaux : le **modèle** et le **méta-modèle**.

**Le modèle** : il correspond à l'abstraction d'un système. La figure 2.1 donne la relation entre un système et un modèle telle qu'elle est définie dans [Bézivin 2005]. Il faut lire sur cette figure : un système est "représenté par" un modèle.

**Le méta-modèle** : il représente la syntaxe utilisée pour définir des modèles. Un méta-modèle est considéré comme un modèle qui fournit les éléments de construction nécessaires pour définir un modèle de niveau inférieur. On dit que tout modèle est "conforme à" son méta-modèle (cf figure 2.1).

#### ILLUSTRATION DES CONCEPTS SUR UN OUVERE-PORTE

Pour illustrer les concepts de modèles et méta-modèles, prenons l'exemple d'un ouvre-porte électrique utilisé pour automatiser le fonctionnement d'un habitat (cf figure 2.2). Il est possible d'en avoir une représentation partielle en modélisant le fonctionnement de ce système pour une porte battante simple (niveau *M1*). On cherche à ce niveau à modéliser les différentes positions

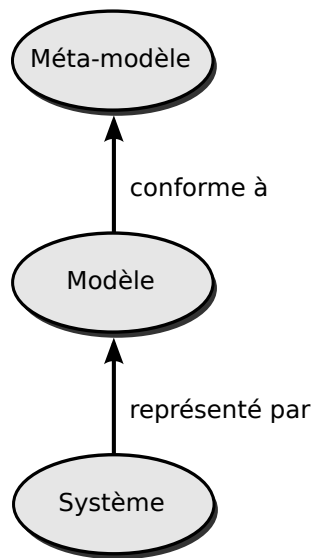


FIGURE 2.1 – Concepts fondamentaux de l’IDM : hiérarchisation des modèles

que peut prendre ce dispositif en fonction des commandes d’ouverture et de fermeture qui peuvent être demandées. Le méta-modèle retenu (niveau *M2*) pour cette représentation est un automate à états finis constitué d’états et de transitions : *FSM (Finite State Machine)*. Ce dernier permet de modéliser les positions de la porte par des états. Ainsi, les états *ouvert* et *fermé* représentent une position que peut prendre cette porte d’un point de vue fonctionnel. Chacun de ces états peut être un état initial du système modélisé. Le changement de positions est représenté par les transitions *ouvrir\_porte* et *fermer\_porte* qui font référence aux commandes sollicitées.

Sur la figure 2.2, on retrouve les niveaux de modélisation introduits précédemment. La spécification *MDA (Model-Driven Architecture)* [Miller et al. 2001] définit un cadre normatif pour séparer explicitement ces niveaux. L’intérêt de l’IDM s’est considérablement accentué lorsque l’*OMG (Object Management Group)* a rendu publique cette spécification. Pour les architectures logicielles et matérielles dirigées par les modèles qui utilisent ce type de standard, il est possible de s’assurer à chaque niveau de modélisation, que les modèles obtenus et réutilisés ont les qualités requises. L’utilisation de modèles pour décrire un système permet également de simplifier sa compréhension, de maîtriser sa complexité et de masquer les détails d’implémentation.

- **M0** : niveau où l’on représente le système. Dans cet exemple, on s’intéresse à un ouvre-porte automatique pour commander des portes battantes.

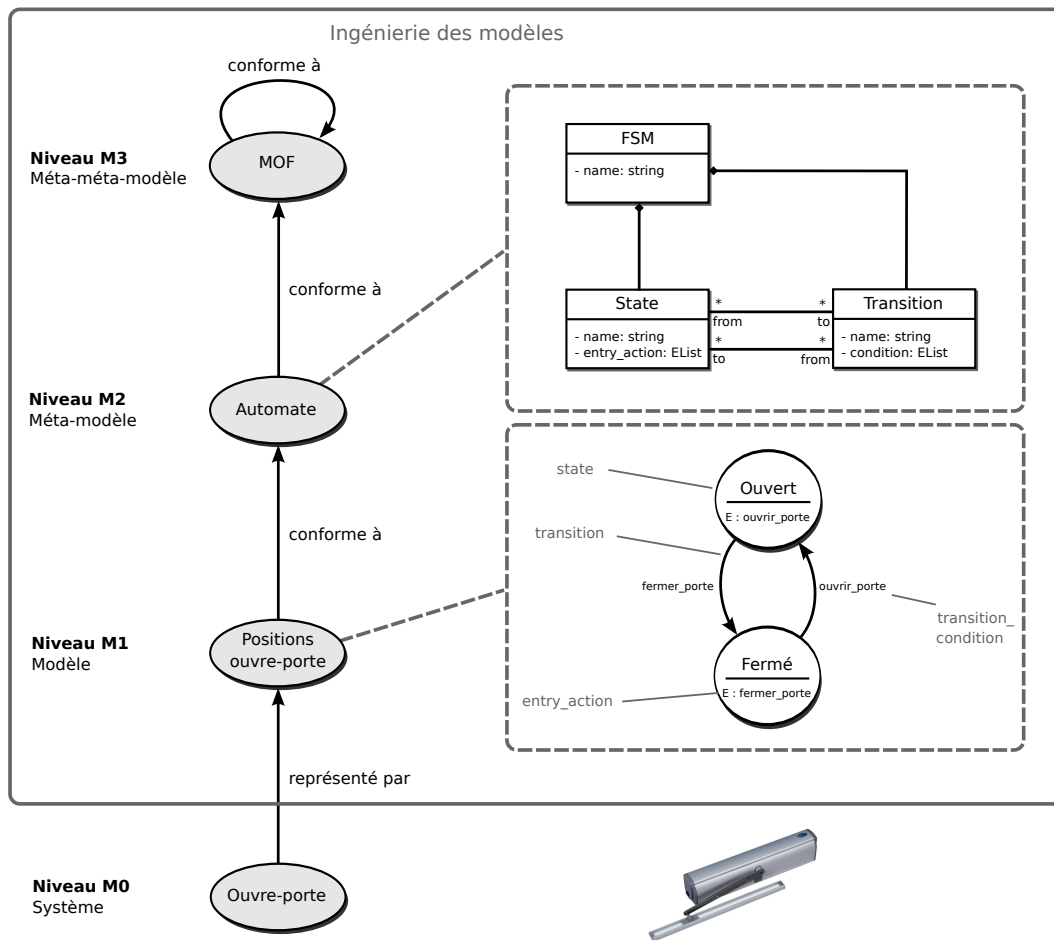


FIGURE 2.2 – Illustration des concepts de l’IDM sur un ouvre-porte automatique

- **M1** : premier niveau de modélisation où l’on définit une représentation partielle du système défini selon un certain point de vue. Le modèle de notre exemple représente les états associés aux différentes positions de l’ouvre-porte.
- **M2** : deuxième niveau de modélisation où l’on définit la syntaxe avec laquelle les modèles doivent être définis. Le méta-modèle de notre exemple est un automate fini. Ainsi tout modèle de niveau *M1* qui représente les positions de l’ouvre-porte doit être défini en utilisant les classes *State* et *Transition*.
- **M3** : un méta-modèle étant considéré comme un modèle, il doit lui aussi être conforme à un méta-modèle. Pour cela, la spécification MDA définit un dernier niveau de modélisation : on parle de méta-méta-modèle qui est conforme à lui-même. Dans l’exemple de la figure 2.2, la description du méta-modèle se fait en utilisant le *MOF (Meta Object Facility)* [MOF 2002] défini par l’OMG.



### 2.1.1.2 Transformation de modèles

La transformation de modèles est au coeur de l'IDM. Elle permet notamment de faire évoluer les modèles *passifs / contemplatifs* comme ceux présentés dans la section précédente, en modèles *actifs / productifs* à partir desquels il est possible de générer l'application cible. Bien que cette approche soit largement répandue dans le milieu de la recherche et commence à l'être dans celui de l'industrie, il n'existe pas de véritable standard pour sa mise en oeuvre. Ainsi, on retrouve différentes propositions de langage de transformation comme *ATL* [Jouault & Kurtev 2006, Jouault *et al.* 2006], *Ker-meta* [Fleurey *et al.* 2006, Falleri *et al.* 2006] ou encore *Smart QVT* [Telecom 2007]. La figure 2.3 montre les principes de base de la transformation de modèles, en suivant le schéma d'application du langage de transformation *ATL* retenu dans ces travaux. À partir du modèle d'entrée *Ma* conforme au méta-modèle *MMa*, la transformation de modèles "*mma2mmb*" permet de générer le modèle *Mb* conforme au méta-modèle *MMb*. Comme le montre la figure 2.3, les règles de transformation *ATL* se définissent au niveau *M2* et s'appliquent ensuite aux modèles du niveau *M1*.

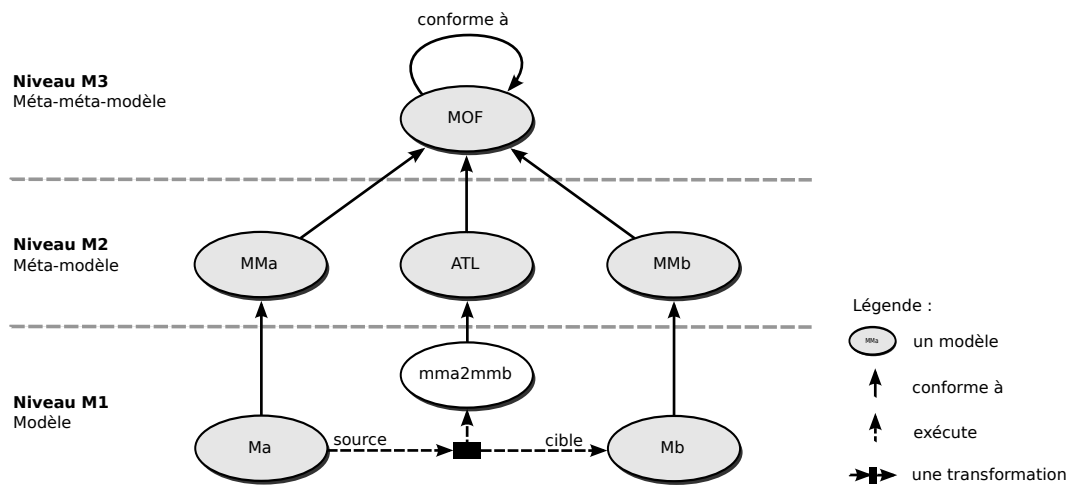


FIGURE 2.3 – Processus de transformation de modèles avec ATL

Suivant les objectifs, on peut aboutir à des schémas d'application divers de ces transformations de modèles, dépassant le cadre de l'illustration présentée sur la figure 2.3. En partant des méta-modèles sources et cibles de la transformation, deux types de transformations [Czarnecki & Helsen 2003, Czarnecki & Helsen 2006] peuvent être identifiés : *endogènes* et *exogènes*. Une transformation est dite endogène si les modèles impliqués sont issus du même méta-modèle. En revanche, lorsque les modèles sources et cibles proviennent de méta-modèles différents, on parle de transformations exogènes. Ces deux catégories peuvent être subdivisées de la manière suivante suivant les objectifs visés :

### Transformation endogène

- **Optimisation** : transformation dont le but est d'améliorer les performances tout en conservant la sémantique
- **Raffinement** : transformation qui opère un changement dans la structure du modèle pour améliorer certains aspects de la qualité du logiciel comme la compréhension, la maintenance, la modularité ou la réutilisation sans en changer le comportement observable
- **Simplification** : transformation appelée aussi *normalisation* dont le but est de réduire la complexité syntaxique du modèle

### Transformation exogène

- **Synthèse** : transformation d'un certain niveau d'abstraction vers un niveau d'abstraction moins élevé (e.g. génération de code)
- **Rétro-ingénierie** : transformation inverse de la synthèse (on cherche à élever le niveau d'abstraction du modèle source)
- **Migration** : transformation d'un programme écrit dans un langage vers un autre langage du même niveau d'abstraction

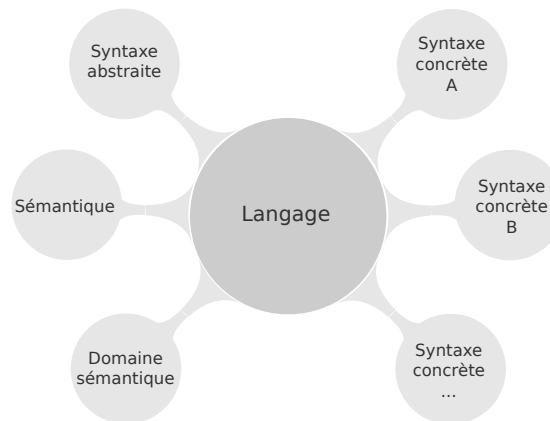
## 2.1.2 Les langages de modélisation

### 2.1.2.1 Définition

Les langages de modélisation se caractérisent tous par leur syntaxe et la sémantique qu'ils véhiculent. Malgré cela, des divergences existent quant à leurs natures, notamment au niveau de leur domaine d'application et de leur expressivité. Contrairement aux langages de modélisation généralistes comme *UML*, les langages de modélisation dédiés sont définis pour répondre aux besoins d'un domaine particulier comme celui de la domotique pour l'assistance à la personne.

**Syntaxe(s)** Un langage de modélisation se définit en premier lieu par sa syntaxe. D'un point de vue théorique, la syntaxe d'un langage de modélisation représente un ensemble comportant tous les mots appartenant à ce langage. Un mot est défini comme un ensemble de symboles qui appartiennent à un alphabet fixé. Ces symboles peuvent être des symboles de type figure géométrique dans le cas des langages graphiques de modélisation. Lorsque l'on définit un langage de modélisation, on distingue généralement sa syntaxe abstraite de sa syntaxe concrète (cf figure 2.4). Cette distinction est d'autant plus vraie quand on s'intéresse aux langages graphiques de modélisation. La syntaxe abstraite spécifie les concepts fondamentaux du langage alors que la syntaxe concrète est vue comme une syntaxe "de surface" [Le Gloahec 2011], celle que l'utilisateur peut manipuler. La syntaxe abstraite est généralement définie par un modèle. La syntaxe concrète s'attache quant à elle uniquement à la forme des modèles (e.g. taille, orientation, couleur, etc.) : elle ne véhicule donc aucune information de structure et de sens. Cette séparation offre deux avantages :

- construire une sémantique sur la base d'une syntaxe abstraite, contenant l'information utile minimale
- permettre d'associer plusieurs syntaxes concrètes à un même langage : on retrouve fréquemment une syntaxe de stockage (e.g. sérialisé en XML) et une syntaxe concrète pour les utilisateurs (plus manipulable, conviviale et esthétique)



Adapté de "Un langage et une plateforme pour la définition et l'exécution de bonnes pratiques de modélisation", V. Le Gloahec

FIGURE 2.4 – Composantes fondamentales d'un langage de modélisation

**Sémantique** Un langage de modélisation se définit aussi par sa sémantique. Elle désigne le lien entre un mot ("le signifiant") et la chose qu'il représente ("le signifié") pour donner un sens à chacune des constructions du langage. Un langage  $L$  est donc défini de manière générale par le tuple  $L = \{S, Sem\}$  [Zhang & Xu 2004], où  $S$  est sa syntaxe et  $Sem$  sa sémantique.

### 2.1.2.2 Les langages dédiés

**Définition** Les langages dédiés (*Domain Specific Language* ou *DSL*) ont la particularité de porter uniquement les concepts d'un domaine précis [Mernik *et al.* 2005]. Ils se distinguent des langages généralistes (*General Purpose Language* ou *GPL*) tels que *UML*, *Java* ou *C++* qui offrent un ensemble de constructions capables de répondre à de nombreux problèmes. En ce sens, les langages dédiés permettent une expression plus directe et plus précise des idées pour des classes plus limitées de problèmes. Dans leur domaine d'application, ils offrent une productivité supérieure à celle des langages généralistes.

Dans le domaine de l'IDM, il est également possible de définir des langages de modélisation dédiés (*Domain Specific Modeling Language* ou *DSML*). Trois approches principales peuvent être retenues pour leur conception [Le Gloahec 2011] :

1. Raffiner un langage généraliste existant en spécialisant certains de ses concepts. Cette approche suppose une proximité importante entre la syntaxe abstraite et la sémantique du langage généraliste et celle du langage dédié. La notion de “profil UML” illustre parfaitement cette approche. Si cette approche a pour avantage de permettre la réutilisation de concepts et outils existants, on peut toutefois lui reprocher son manque d’expressivité lorsque l’on s’intéresse à un domaine spécifique [Pardillo & Cachero 2010].
2. Étendre ou restreindre un langage existant avec des concepts et structures propres au domaine visé. Plus flexible que l’approche précédente, elle permet de réduire les efforts d’apprentissage, à défaut de pouvoir réutiliser des outils existants. *SysML*, dédié au domaine de l’*Ingénierie Système* est représentatif de cette approche [SysML 2008, Friedenthal *et al.* 2011].
3. Définir un nouveau langage dédié “*from scratch*”. Cette approche offre une totale flexibilité et un pouvoir d’expression important en adéquation avec les problèmes d’un domaine particulier auxquels le langage doit répondre. Le développement d’outils devient alors obligatoire. Pour pallier ce problème, de nombreux méta-outils permettent de faciliter la définition des *DSML* et la génération d’outils associés. Parmi ces méta-outils, on peut citer :
  - *Eclipse EMF/GMF* [Steinberg *et al.* 2008, Gronback 2009]
  - *GME* [Davis 2003]
  - *Microsoft DSL Tools* [Greenfield & Short 2003]
  - *Kermeta* [Muller *et al.* 2005]
  - *AM3/KM3* [Jouault & Bézivin 2006]

### 2.1.2.3 Langage et MDA

Le langage dans lequel un modèle est exprimé doit être rigoureusement défini pour être productif. En réponse à ce besoin, l’OMG introduit la notion de métamodèle (cf section 2.1.1.1) de la manière suivante :

#### DEFINITION D’UN META-MODELE

“Un méta-modèle est un modèle qui définit le langage d’expression d’un modèle” Meta Object Facility [OMG 2006]

Un méta-modèle est donc vu comme un modèle d’un langage de modélisation. Tout comme le lien “représenté par” relie un modèle au système modélisé, la relation “conforme à” caractérise le lien entre un modèle et le métamodèle auquel il se conforme [Bézivin 2004]. Dans cette dernière relation, le méta-modèle joue le rôle de modèle de langage, et le langage joue le rôle de système modélisé (cf figure 2.5).

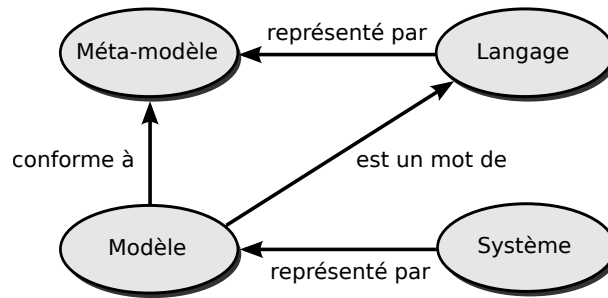


FIGURE 2.5 – Relations entre système, modèle, méta-modèle et langage




### ILLUSTRATION EN CARTOGRAPHIE

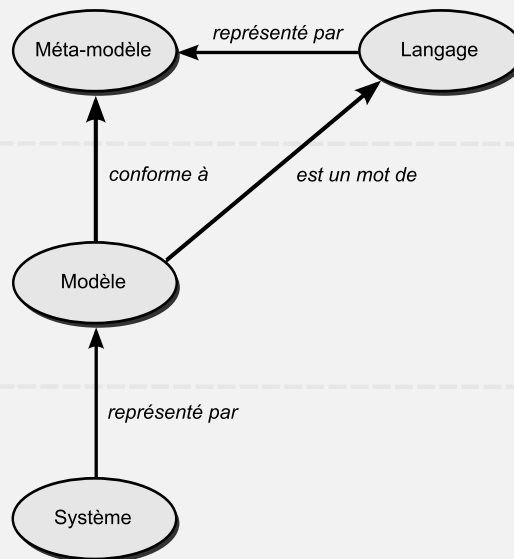
Un exemple repris dans de nombreux travaux sur l'IDM [Favre *et al.* 2006] permet d'illustrer ces relations. Dans le domaine de la cartographie, il est indispensable d'associer à chaque carte une légende qui décrit le langage utilisé pour décrire cette carte. La carte (i.e. le modèle) doit être conforme à cette légende (i.e. son métamodèle). Dans cette illustration, le méta-modèle est une représentation du "langage des cartes IGN".



Le "langage des cartes IGN" : l'ensemble des cartes décrites selon les conventions IGN, en tant que mode d'expression/communication d'information.

#### Légende

-  Ligne de coupe, sentier...
-  Etablissement hospitalier
-  Camping
- ...



## 2.2 État de l'art & positionnement

Différents travaux ont porté sur 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 2.2.1). Ce sont les 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. Parmi ces outils, on peut distinguer les langages de modélisation généralistes comme *UML (General Purpose Langage, section 2.2.2)* et les langages de modélisation dédiés (*Domain Specific Modeling Langage, section 2.2.3*).

### 2.2.1 Modélisation spécifique à une plateforme

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* et *LonMaker* sont deux des outils commerciaux les plus connus [KNX Association 1993, Echelon Corporation 1998], 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.

### 2.2.2 Modélisation généraliste indépendante

Les propositions basées sur l'IDM permettent au concepteur de capturer des spécifications indépendamment de la plateforme cible (*Platform Independent Modeling*) en s'abstrayant des spécificités technologiques. L'objectif principal est de garantir l'interopérabilité des systèmes électroniques/informatiques dans un environnement domotique complet.

*PerovML* [Muñoz & Pelechano 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'interfaces 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 spécifier les caractéristiques matérielles et logicielles 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'interopérabilité entre systèmes domotiques hétérogènes utilisant différents protocoles de communication. L'objectif de ce middleware est de garantir l'interconnexion des réseaux et l'ouverture à des protocoles de haut niveau, de favoriser les accès nomades, d'offrir une solution dynamique de déploiement et de 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 de la domotique pour l'assistance à la personne. Pour le concepteur non-expert en informatique, les modèles issus de ce type de langage sont complexes à manipuler, nécessitant un bagage technique avancé.

### 2.2.3 Modélisation de domaine indépendante

Les langages de modélisation spécifiques à un domaine particulier (*Domain Specific Modeling Language*) permettent de prendre en compte exclusivement les besoins précis d'un domaine, tout en s'abstrayant des spécificités matérielles.

*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.

Dans [Reinisch *et al.* 2009, Bonino *et al.* 2009], une ontologie est proposée pour modéliser les systèmes domotiques, permettant d'en garantir l'interopérabilité. Nous pensons que l'IDM apporte des solutions pertinentes pour cela de part les outils existants

qui permettent la création rapide d'éditeurs graphiques, la mise en place de transformation de modèles, de génération de texte et la vérification syntaxique de modèles (cf section 4.1.3 du chapitre 4).

Même si de nombreux efforts ont été faits pour faciliter l'appréhension de la conception, les langages de modélisation présentés ci-dessus adoptent 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 sont 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) : il n'est par exemple pas possible d'exprimer une interdiction sur son environnement ou encore de prendre en compte les aléas qui peuvent survenir au niveau des équipements domotiques.
- le concepteur doit être une personne sensibilisée, voire experte en domotique pour utiliser ce langage de modélisation.

L'originalité des travaux présentés dans ce mémoire réside dans la prise en compte de ces deux points (cf tableau 2.1) :

|                     | Multi-plateformes | Spécifique au domaine<br>(de la domotique) | Expressivité | Accessibilité<br>(au non-expert) |
|---------------------|-------------------|--|--------------|----------------------------------|
| [LonMaker, 98] *    | ✗                 | ✓  | ✗            | ✗                                |
| [ETS, 01] *         | ✗                 | ✓  | ✗            | ✗                                |
| [PervML, 07] **     | ✓                 | ✗  | ✓            | ✗                                |
| [EntiMid, 08] **    | ✓                 | ✗  | ✓            | ✗                                |
| [DogONT, 08] ***    | ✓                 | ✓  | ✓✓           | ✓                                |
| [useML, 11] **      | ✓                 | ✗  | ✓            | ✗                                |
| [HomeDSL, 11] ***   | ✓                 | ✓  | ✓✓           | ✓                                |
| [IntelHome, 12] *** | ✓                 | ✓  | ✓✓✓          | ✓✓✓                              |

\* Platform Specific Modeling

\*\* Platform Independent Modeling (Unified Modeling Language)

\*\*\* Platform Independent Modeling (Domain Specific Modeling Language)

TABLE 2.1 – Langages de modélisation pour la conception domotique

## 2.2.4 Choix retenu

La modélisation indépendante de toute plateforme, répondant toutefois aux besoins spécifiques de la domotique pour l'aide à la personne nécessite la définition d'un DSML. Conformément au processus de conception centré utilisateur présenté sur la figure 1.8 du chapitre 1, notre choix s'oriente vers une représentation multi-vues. Le



DSML défini comporte ainsi deux vues mises en oeuvre par deux modèles distincts : une vue physique pour garantir l'étape de spécification du contexte d'utilisation (modèle d'environnement) et une vue fonctionnelle pour l'étape de spécification des besoins utilisateur (modèle d'interactions).

La vue environnement représente la dimension architecturale du système domotique étudié. Elle permet de décrire la structuration du système en sous-systèmes et équipements. La vue interactions représente quant à elle la dimension fonctionnelle pour préciser le fonctionnement du système. Cette vue est essentielle pour personnaliser / adapter le système aux besoins de l'utilisateur. La définition de la vue physique et de la vue fonctionnelle nous a semblé pertinente pour réduire la complexité de conception et pour mieux guider le concepteur non-expert. Cette distinction permet de projeter différentes facettes du système sur des dimensions identifiées et pertinentes en fonction des métiers et des préoccupations des différents intervenants. L'intervention d'un ergothérapeute ou de toute autre personne proche des besoins de l'utilisateur est par exemple justifiée pour l'adaptation de services domotiques en fonction des capacités motrices / gestuelles de ce dernier (dimension fonctionnelle). L'ingénieur ou domoticien peut quant à lui intervenir au niveau de la définition de l'environnement domotique pour positionner les équipements en tenant compte de la portée des protocoles de communication utilisés (dimension physique)<sup>1</sup>.

## 2.3 Description du DSML

Le langage de modélisation retenu pour la description de systèmes domotiques d'assistance comprend deux vues :

1. Une vue *Environnement* pour la modélisation de l'environnement domotique basée sur la notion de composants logiciels définis à des niveaux d'abstraction différents
2. Une vue *Interaction* pour la modélisation des interactions entre l'utilisateur et son système domotique en fonction des besoins de ce dernier

Ces deux vues permettent de séparer explicitement l'architecture physique du système domotique basée sur les composants, de l'architecture logique orientée services. Les concepts présents dans ces deux vues sont définis et regroupés au sein de deux méta-modèles présentés dans les sections 2.3.1 et 2.3.2.

### 2.3.1 Méta-modèle de l'environnement

#### 2.3.1.1 Modélisation de l'environnement par composants

Par environnement domotique, nous entendons un logement équipé en domotique qu'un ou plusieurs utilisateurs peuvent occuper. Qu'il soit défini à l'échelle d'un ap-

---

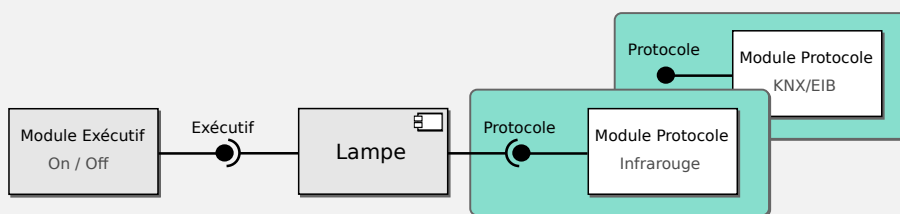
1. Notre objectif dans ces travaux est malgré tout de permettre à une personne non-experte d'intervenir dans cette étape de la conception.

partement 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), d'éventuels capteurs et interfaces de commande (mobiles ou murales), de serveurs, de bus domotiques, de connexions de natures différentes suivant le type de protocole utilisé, etc. Pour représenter les équipements domotiques et les espaces physiques, nous introduisons une approche composant pour le domaine de la domotique, qui respecte la définition issue du paradigme CBSE (*Component Based Software Engineering*) [Szyperski *et al.* 2002]. Un composant est ainsi défini comme une "boîte noire" qui fournit uniquement ses interfaces et ses exigences. Il est considéré comme substituable, composable et réutilisable :

- **Substituable** : un composant peut être remplacé par un autre composant à partir du moment où ce dernier possède la même interface.
- **Composable** : un composant peut être assemblé avec d'autres composants pour former des composants de plus haut niveau et aboutir ainsi à des fonctionnalités plus complexes.
- **Réutilisable** : un composant peut être stocké dans une bibliothèque en vue d'être réutilisé pour une autre application.

### SUBSTITUTION

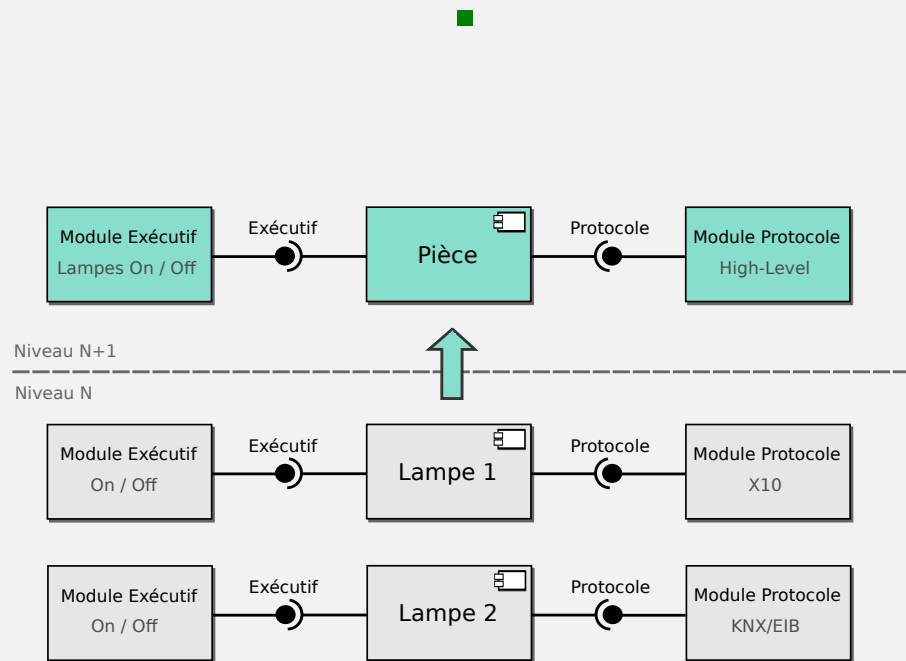
Il est possible par exemple de changer une lampe contrôlable en infrarouge par une lampe contrôlable par un bus domotique, à partir du moment où elle présente les mêmes fonctionnalités. Sur la figure ci-dessous, un composant *Lampe* peut avoir deux types de protocole de communication (*Infrarouge* ou *KNX/EIB*) pour le même module exécutif (opérations *On/Off*).



### COMPOSITION

À partir des composants *Lampe 1* et *Lampe 2* (cf figure ci-dessous), le composant *Pièce* peut être défini. Ce composant de haut niveau se voit associer un module exécutif avec les opérations *On/Off* qui s'appliquent indirectement

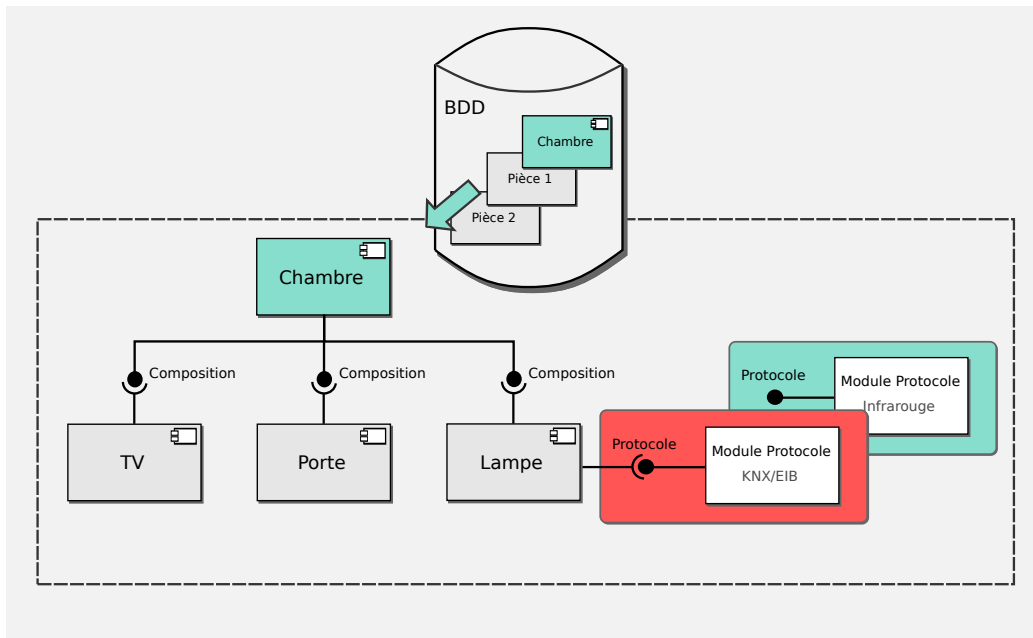
sur les deux lampes *Lampe 1* et *Lampe 2*. En revanche, ce type de composant ne requiert pas de protocole de communication particulier.



### REUTILISATION

Les chambres d'un centre de soins présentent souvent le même environnement domotique. Dès lors, après avoir modélisé une chambre à partir de composants élémentaires, celle-ci peut être stockée et réutilisée, en conservant certains paramètres et en en modifiant d'autres. Sur la figure ci-dessous, le composant *Chambre* est extrait d'une bibliothèque de composants, puis adapté pour intégrer une lampe contrôlable en infrarouge plutôt que par le bus domotique *KNX/EIB*.





Dans ces travaux, à chaque composant logiciel est associé un composant physique. 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.

Le patron de conception (*design pattern*) "Composite" se retrouve ainsi au coeur de la modélisation de l'environnement domotique (cf figure 2.6). Un composant peut être élémentaire (*Basic\_Component* : lumières, portes, volets, etc) ou bien agrégé (*Aggregate\_Component*) à l'échelle d'une pièce. 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 : il est composé d'un nom (*name*), et d'un identifiant unique (*id*).

- Un composant élémentaire doit de plus être renseigné sur le protocole de communication utilisé : *protocol* (*Bluetooth*, *KNX/EIB*, *Infrared*, etc.). Il s'agit de la seule information d'implémentation à renseigner pour le concepteur à ce niveau.
- Les composants agrégés qui peuvent être définis à l'échelle d'une pièce sont associés entre eux par l'intermédiaire du composant élémentaire de type *Porte*. Une porte est donc vue comme une interface entre deux pièces (cf figure 2.7). Cette modélisation peut être utile pour repérer où se trouve l'utilisateur en fonction de l'utilisation de la porte (cf chapitre 4).

Chaque composant, élémentaire ou agrégé, se voit associer une ou plusieurs opérations (classe **Operation**). Par exemple, le composant *Lampe* possède les opérations *Allumer\_Lumière* et *Éteindre\_Lumière*. Ces opérations sont par la suite liées à des comman-

FIGURE 2.6 – Extrait du méta-modèle *Environnement*

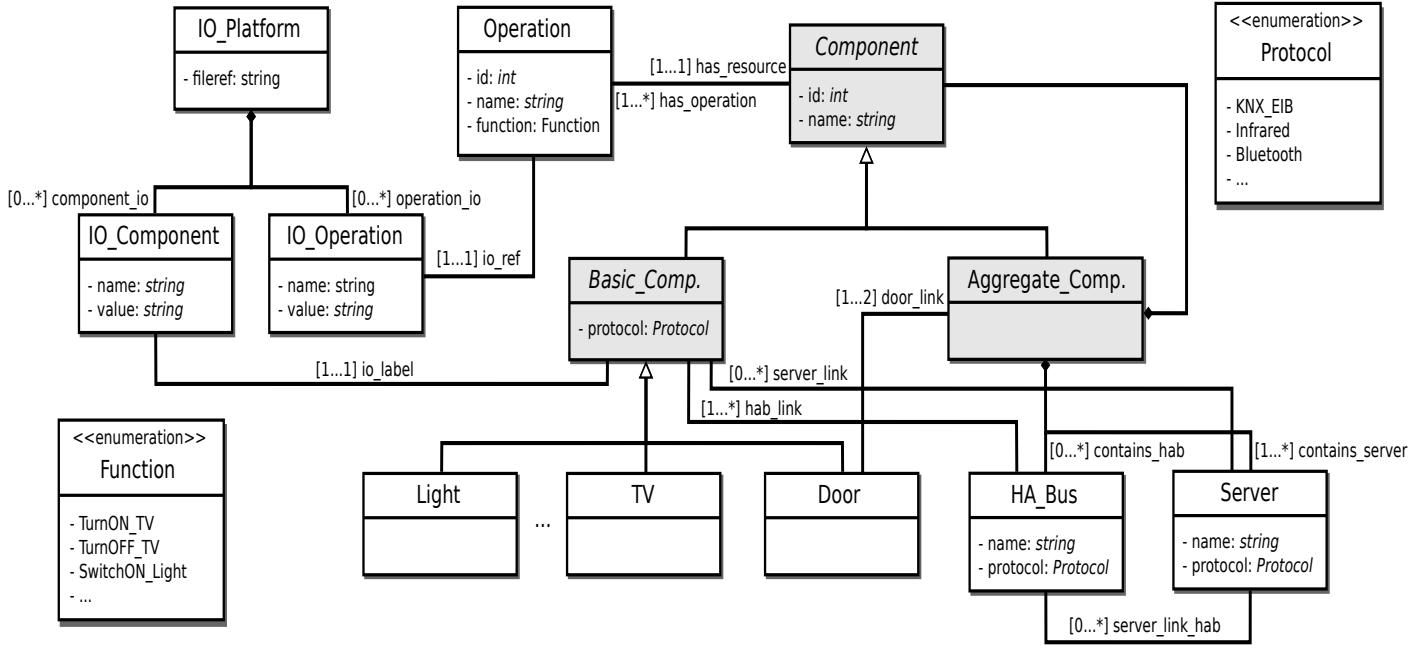


FIGURE 2.7 – Composant *Porte* vu comme une interface entre deux pièces

des physiques (par exemple, des commandes infrarouges) appelées **IO\_Operation** issues d'un fichier de configuration (**fileref** de la classe **IO\_Platform**). Ce fichier de configuration a été défini par les domoticiens lors de l'installation et l'intégration du bus domotique et des différents équipements. Il fait le lien entre les commandes physiques et les équipements présents dans l'habitat.

Pour commander ces équipements, il est possible de modéliser des bornes domotiques, définies dans des composants agrégés. Il s'agit de cartes électroniques qui peuvent commander les équipements de différentes façons : soit directement en utilisant un protocole de communication sans fils (**server\_link**), soit en passant par un bus domotique pour centraliser la commande de tous les équipements (**server\_link\_hab**).

### 2.3.1.2 Une approche ascendante de conception

La méthodologie de conception indépendante de toute informatisation prône une approche ascendante (*bottom-up*) de conception. À partir de composants élémentaires (représentants des équipements domotiques), 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. 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.

La proposition d'une approche ascendante de conception à un non-expert peut susciter des questions, une approche descendante semblant plus adaptée à l'absence de compétences techniques 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. Malgré tout, si la méthodologie préconise une approche *bottom-up* indépendamment de toute informatisation, la modélisation retenue laisse la possibilité d'adopter une approche descendante (*top-down*) lorsqu'il s'agit d'utiliser l'outil développé. Une fois la modélisation de

l'environnement en tête, il peut être plus intuitif pour le concepteur non-expert de partir du composant représentant le système global pour par la suite raffiner ce modèle jusqu'à la définition de composants élémentaires auxquels on doit associer différents protocoles de communication.

Le modèle d'environnement que nous proposons fait la distinction entre le point de vue Ressource et le point de vue Système (cf tableau 2.2) :

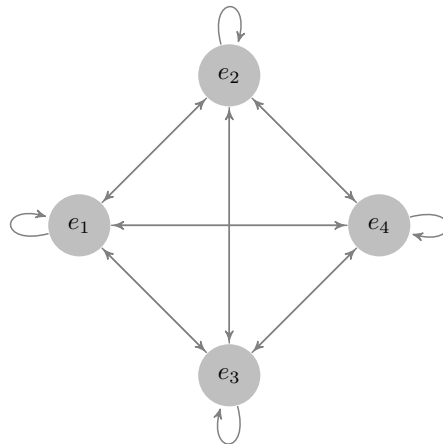
| ENVIRONNEMENT |            | INTERACTIONS        |
|---------------|------------|---------------------|
| Ressource     | Système    |                     |
| Opération     | Fonction   | Service : "Je veux" |
| État          | Super-état | Mode : "J'interdis" |

TABLE 2.2 – Vues Environnement et Interactions et concepts associés

**Point de vue Ressource :** selon ce point de vue, une **opération** permet de changer les **états** d'une ressource particulière. Une "ressource" correspond à 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* qui permettent de changer l'état de cette ressource. La figure 2.8 montre le fonctionnement d'une ressource *R1*. Cette ressource possède quatre états représentés par les noeuds du graphe :  $e_1$ ,  $e_2$ ,  $e_3$  et  $e_4$ . Les opérations sont représentées par les arcs et permettent d'aboutir à n'importe quel état depuis n'importe quel état. Il s'agit donc d'un graphe complet où chaque noeud représente l'état souhaité par l'utilisateur (on ne s'intéresse pas aux états transitoires).

**Point de vue Système :** défini pour élever le niveau d'abstraction, ce point de vue est responsable de la supervision du système domotique. Selon ce point de vue, une **fonction** est définie par un concept abstrait qui, lorsqu'il est implémenté par une ressource, fournit une opération. Une fonction peut également être implémentée par plusieurs ressources. Par exemple, le fonction *Allumer la lumière* peut être implémentée sur la ressource *Lampe 1* pour former l'opération *Allumer la lampe 1*. De la même manière, nous définissons un **super-état** comme un concept abstrait, qui lorsqu'il est implémenté par une ressource, fournit un état.

La définition de ces concepts fait référence aux travaux issus des Systèmes Automatisés de Production (SAP). Dans ce contexte, la distinction entre opérations et fonctions [Toguyéni *et al.* 2003] est utilisée pour la reconfiguration en cas de panne de ce genre de systèmes. Lorsqu'une opération échoue à cause d'une panne sur une ressource la



$$E_{Global}(R1) = \{e_1, e_2, e_3, e_4\}$$

FIGURE 2.8 – Représentation graphique du fonctionnement d’une ressource  $R1$

supportant, un système de reconfiguration permet de garantir une qualité de services minimum en activant la même fonction sur une autre ressource. (Cette distinction est également intégrée dans un DSL pour les systèmes de production manufacturiers flexibles [De Lamotte *et al.* 2007].) Nous proposons alors d’étendre cette définition pour intégrer explicitement les concepts *états* et *super-états*. La figure 2.9 illustre cette extension sur un exemple de système domotique composé, entre autres, de deux ressources  $R1$  et  $R2$ .  $R1$  est une porte standart comprenant les opérations *Open R1* ( $Op_{R1-f1}$ ) et *Close R1* ( $Op_{R1-f2}$ ).  $R2$  est une porte coulissante comprenant les opérations *Open R2* ( $Op_{R2-f1}$ ), *Close R2* ( $Op_{R2-f2}$ ) et *Half-Open R2* ( $Op_{R2-f3}$ ). Si l’on considère la fonction *Close the Door*( $f2$ ), on s’aperçoit que cette dernière peut être mise en oeuvre par les ressources  $R1$  et  $R2$ . Il en est de même pour le super-état associé *Door Closed* ( $S_{st-f2}$ ). On peut également vérifier que chaque opération est associée à un état.

À la différence des SAP, les systèmes domotiques d’assistance sont avant tout dédiés à la personne. C’est pourquoi nous introduisons une deuxième vue dans le but de séparer la modélisation physique basée sur les composants de la modélisation logique considérant les besoins spécifiques de l’utilisateur. Les opérations et les fonctions sont exprimées en terme de **services**, alors que les interdictions sont exprimées en terme de **modes**. 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 définies à l’aide du modèle d’interactions dont le méta-modèle est détaillé dans la section suivante.



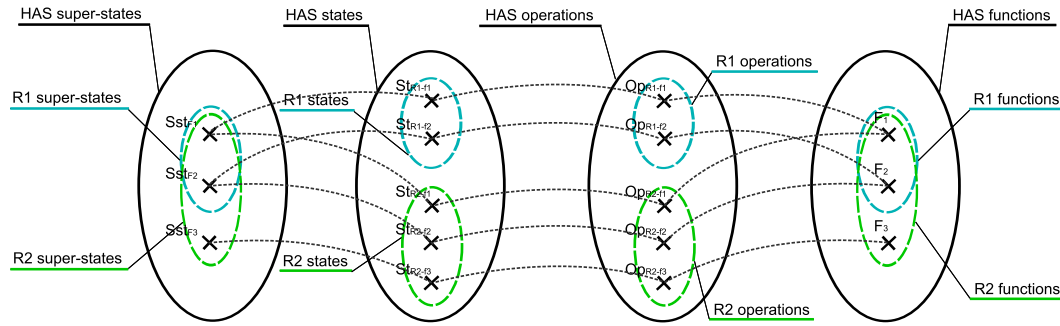


FIGURE 2.9 – Correspondance entre états, super-états, opérations et fonctions

## 2.3.2 Méta-modèle des interactions

Dans cette vue fonctionnelle du *DSML*, nous considérons la manière dont l'utilisateur doit pouvoir interagir avec son environnement. Notre objectif est 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 (section 2.3.2.1). De manière complémentaire, la seconde permet d'exprimer la formule "J'interdis" mise en oeuvre par les modes d'interdiction (section 2.3.2.2).

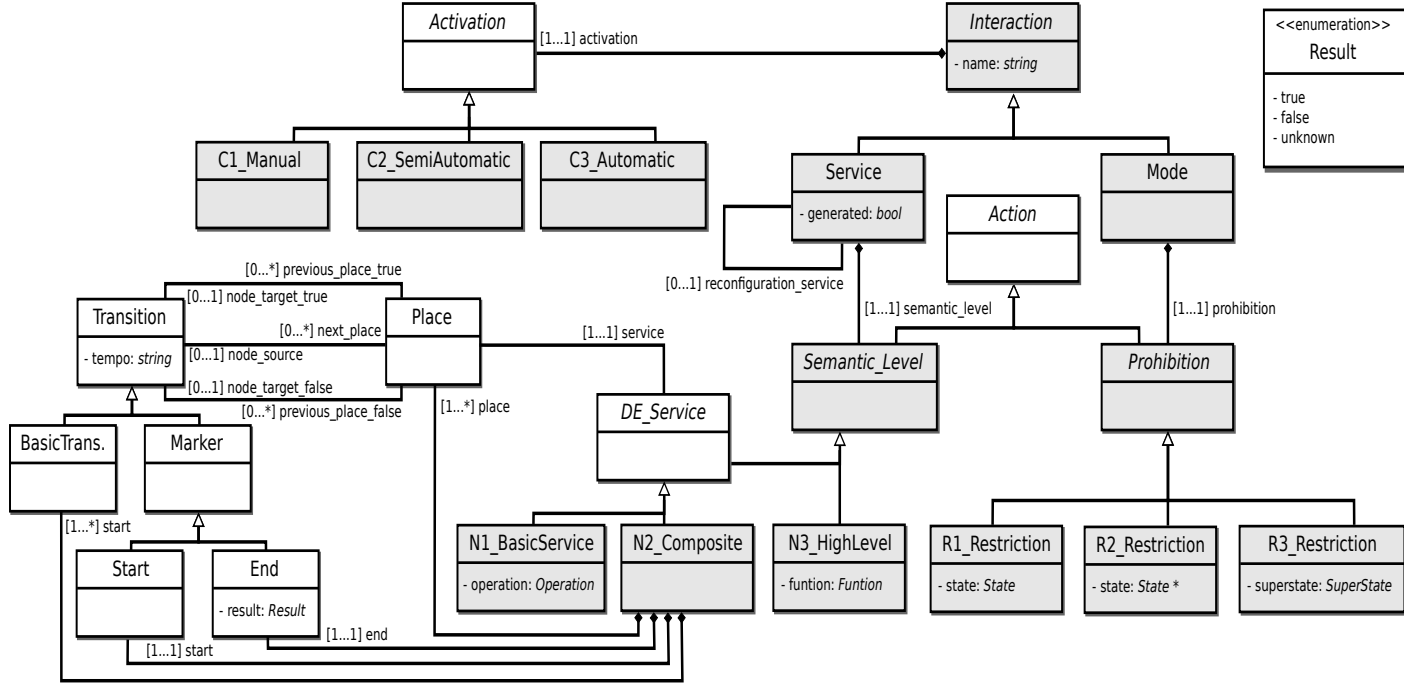
### 2.3.2.1 Les services

Les services sont définis pour prendre en compte la diversité des besoins que les personnes dépendantes, notamment les personnes à mobilité réduite, peuvent exprimer [De Lamotte *et al.* 2008]. Le modèle de services présenté dans cette section permet d'exprimer des requêtes sur des ressources domotiques indépendamment de l'interface utilisée et d'automatiser le fonctionnement d'un habitat.

Pour cela, nous définissons un service domotique 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 définir un service selon le niveau sémantique qu'il véhicule :

- $N1$  : service élémentaire associé à une opération
- $N2$  : scénario composé de plusieurs services  $N1$
- $N3$  : service de haut niveau sémantique associé à une fonction

FIGURE 2.10 – Extrait du méta-modèle *Interaction*



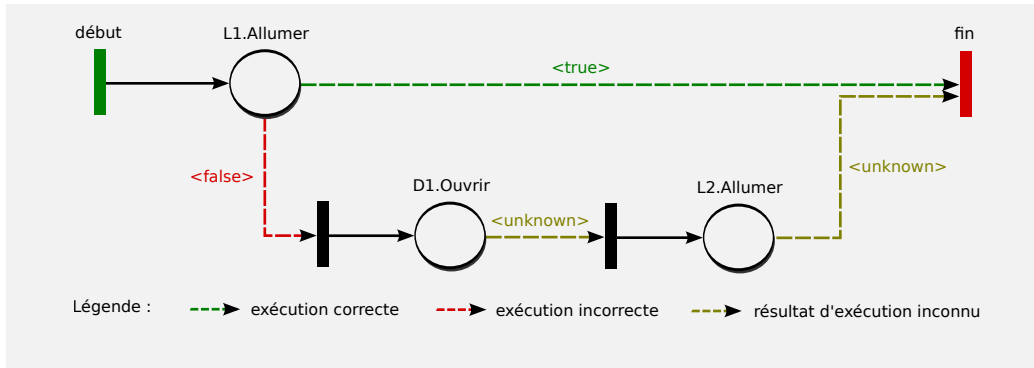
**service N1** La figure 2.10 présente un extrait du méta-modèle retenu pour définir les interactions utilisateur-système. On y retrouve les services définis en fonction du niveau sémantique qu'ils véhiculent (*N1\_BasicService*, *N2\_Composite* et *N3\_HighLevel*). Les services de niveau sémantique *N1* sont directement associés à une opération d'une ressource. Si on prend l'exemple de la *Lampe 1*, *allumer la lampe 1* peut être considéré comme un service de type *N1*.

**service N2** La définition de scénarios (*N2\_Composite*) consiste à définir un graphe orienté avec des places, des transitions et des arcs. La représentation graphique choisie est celle des *Réseaux de Petri* (RdP) [Petri 1962], un modèle mathématique couramment utilisé pour représenter les systèmes à événements discrets. Chaque place est associée à l'activation d'un service domotique à événements discrets (services élémentaires ou scénarios). Les transitions, qui doivent être définies entre chaque place, peuvent être de différents types : une transition classique (*Basic*), un marqueur de début (*Start*) ou un marqueur de fin (*End*). Elles peuvent se voir associer une temporisation (*tempo*) permettant de définir des temps d'attente avec l'activation de certains services. Les arcs permettent de lier les places aux transitions et vice versa. Le concepteur peut choisir un type d'arc suivant le résultat d'exécution (*true*, *false* ou *unknown*) de chaque service et définir ainsi des services alternatifs pour garantir la sûreté de fonctionnement du système domotique. Le graphe résultant peut alors être interprété à l'exécution en considérant les chemins alternatifs possibles (ainsi que l'activation de services alternatifs).

#### SCENARIO "ALLUMER LAMPE"

Considérons deux pièces *P1* et *P2* dans lesquelles se trouvent respectivement les lampes *L1* et *L2*. La porte *D1* sépare ces deux pièces. Le scénario "*Allumer Lampe*" représenté sur la figure ci-dessous permet d'allumer la lampe *L1* lorsque l'utilisateur se trouve dans la pièce *P1*. Si l'exécution de ce service s'est correctement réalisée, le scénario s'arrête. Sinon, le système ouvre la porte et allume la lampe *L2* pour assurer une qualité de services minimum. Il s'agit de la représentation adoptée dans ces travaux pour définir des scénarios : elle reprend les éléments graphiques des RdP sans leurs sémantiques (e.g. sans le système de jetons).





**service N3** Les services de haut niveau sémantique sont directement associés à une fonction. Prenons l'exemple d'une pièce *Salon*, dans laquelle se trouvent deux lampes *L1* et *L2*. *Allumer la lumière* peut être considéré comme un service de type *N3* qui peut s'appliquer sur une des deux lampes. Le choix de l'activation d'une lampe plutôt qu'une autre est fait après analyse des facteurs contextuels (e.g. préférences de l'utilisateur, sa localisation). La résolution de ce genre de services dépasse le cadre de cette thèse. Malgré tout, nous pensons que l'exploitation d'un raisonnement logique basé sur les ontologies et les outils du web sémantique permettrait d'y apporter des solutions (cf chapitre 4 et chapitre 6).

### 2.3.2.2 Les modes

Pour compléter le pouvoir d'expression du DSML, nous définissons un *mode* comme une restriction sur l'espace d'états d'une ressource. Le système domotique étant considéré comme un système à événements discrets, le fonctionnement de chaque ressource le constituant peut être représenté par un graphe d'états-transitions. Considérons par exemple la ressource *R1*, dont les états (et opérations associées) sont représentés sur la figure 2.8 par un graphe complet. Cette ressource comprend l'espace d'états  $E_{Global}(R1) = \{e_1, e_2, e_3, e_4\}$  où chaque état est potentiellement accessible à partir de n'importe quel état. Les modes visent à limiter temporairement cet espace d'états.

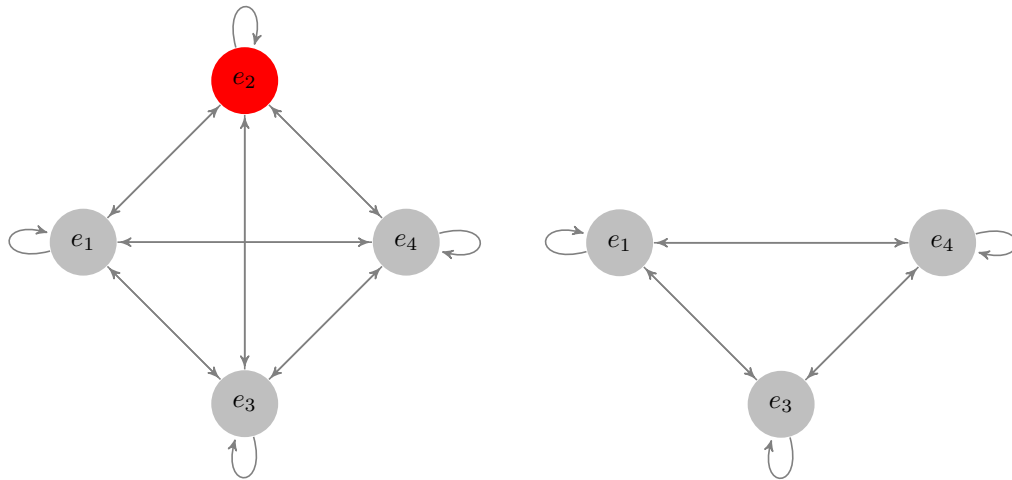
Chaque mode (vue *interactions*) est interprété par un espace d'évolution  $E_r$  (vue *environnement / ressource*). À travers les modes, le concepteur a la possibilité de restreindre l'espace d'états global  $E_{Global}(r) = \{e_1, \dots, e_N\}$  d'une ressource  $r$ , en interdisant un ensemble d'états :

$$\begin{aligned} E_{Interdit}(r) &= \{e'_1, \dots, e'_M\}, \quad M < N \\ E_{Interdit}(r) &\subset E_{Global}(r) \end{aligned} \quad (2.1)$$

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

$$E_{Access}(r) = E_{Global}(r) \setminus E_{Interdit}(r) \quad (2.2)$$

La ressource  $R1$  introduite précédemment peut par exemple se voir interdire l'état  $e_2$  lors de l'activation d'un mode (cf figure 2.11). Cette interdiction est définie du point de vue de l'utilisateur (cf graphe d'états à gauche de la figure). Ce mode se traduit du point de vue ressource par un espace d'états restreint  $E_{Access}(R1)$  (à droite de la figure) comprenant les états potentiellement accessibles  $e_1, e_3$  et  $e_4$ .



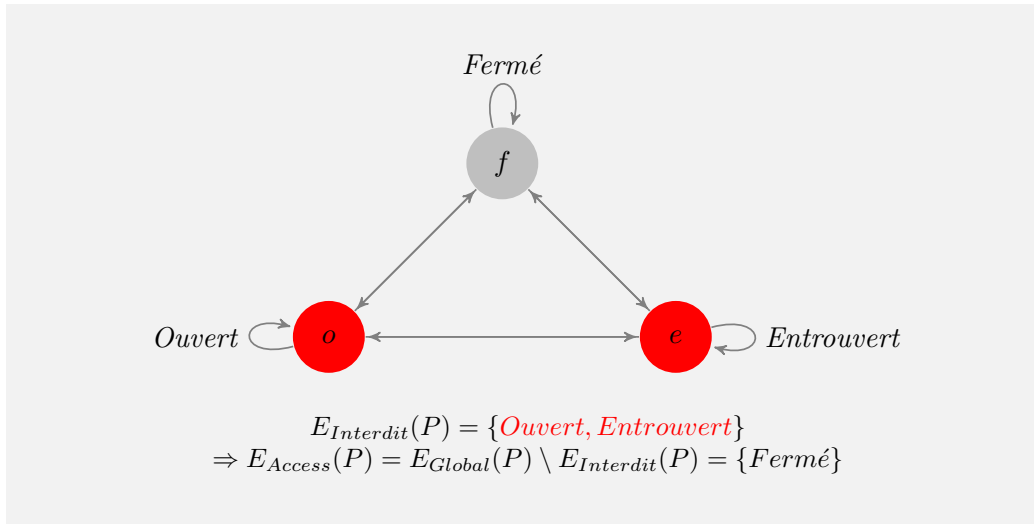
$$E_{Interdit}(R1) = \{e_2\} \Rightarrow E_{Access}(R1) = E_{Global}(R1) \setminus E_{Interdit}(R1) = \{e_1, e_2, e_3, e_4\} \setminus \{e_2\} = \{e_1, e_3, e_4\}$$

FIGURE 2.11 – Activation d'un mode d'interdiction sur la ressource  $R1$

### MODE "ANTI-FUGUE"

Pour illustrer la définition de modes d'interdiction, prenons l'exemple d'une porte coulissante  $P$  qui comporte les états  $E_{Global}(P) = \{Ouv\text{ert}, \text{Entrouv\text{ert}}, \text{Ferm}\acute{e}\}$  (cf figure ci-dessous). Le mode "Anti-Fugue" est défini pour éviter que des patients atteints d'une démence de type Alzheimer à un stade sévère ne puissent sortir de la chambre pendant la nuit. Pour prévenir d'éventuelles fugues, on interdit l'ensemble d'états  $E_{Interdit}(P) = \{Ouv\text{ert}, \text{Entrouv\text{ert}}\}$  (point de vue *utilisateur*) pendant cette période. En conséquence, seul l'état *Fermé* peut être actif : les services *Ouvrir porte* et *Entrouvrir porte* ne sont plus disponibles.



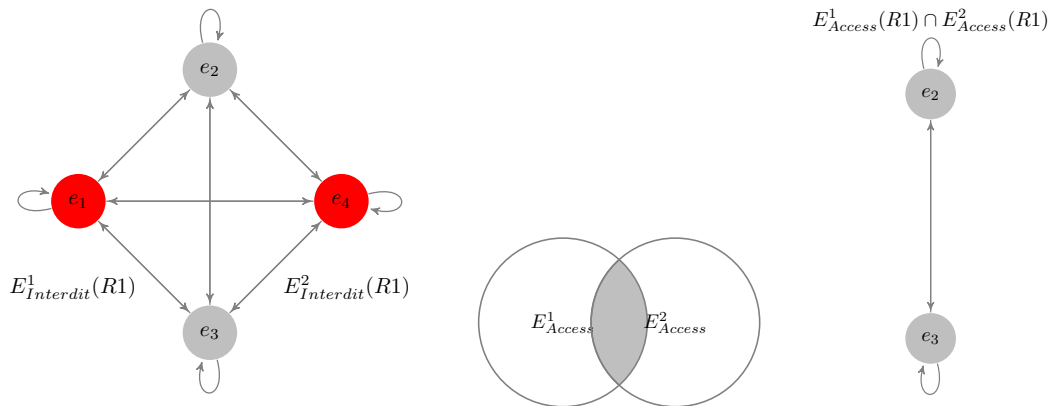


Par ailleurs, plusieurs restrictions sur l'espace d'états d'une ou plusieurs ressources peuvent être activées simultanément, par l'intermédiaire d'autant de modes. Cela permet de contraindre de manière encore plus précise l'espace d'évolution d'une ou plusieurs ressources. Par exemple, si  $E_{Access}^1 r$  représente l'espace d'états autorisés par un premier mode (par exemple *économie d'énergie*), et que  $E_{Access}^2 r$  représente l'espace d'états autorisés par un second mode (par exemple *enfant sans surveillance*), alors, l'espace des états autorisés par ces deux modes sur la ressource  $r$  se calcule par intersection de  $E_{Access}^1 r$  et de  $E_{Access}^2 r$  :

$$E_{Access}(r) = E_{Global}(r) \setminus \bigcup_{i=0}^k E_{Interdit}^i(r) = \bigcap_{i=0}^k E_{Access}^i(r) \quad (2.3)$$

La figure 2.12 montre l'activation de deux modes sur la ressource  $R1$ . Le premier mode interdit l'état  $e_1$ . Le deuxième interdit l'état  $e_4$ . L'intersection des deux espaces d'évolution associés à chacun de ces modes définit l'espace d'évolution final de la ressource  $R1$ . Dans cet exemple, la ressource  $R1$  pourra atteindre les états  $e_2$  et  $e_3$  lors de l'activation de ces deux modes.

La définition des modes ne se limite pas seulement à des contraintes sur les ressources. Un gestionnaire permet de garantir une certaine cohérence des 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 autorisée que si la ressource a au moins un état potentiellement

FIGURE 2.12 – Représentation graphique de l'activation de deux modes sur  $R1$ 

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.

De manière complémentaire aux services, trois niveaux de restriction sont définis :

- $R1$  : restriction sur un ou plusieurs états d'une ou plusieurs ressources (composants élémentaires uniquement).
- $R2$  : restriction sur un ou plusieurs états d'une ou plusieurs ressources (composants élémentaires et composants agrégés).
- $R3$  : restriction indépendante de toute ressource. Ce type de restriction permet l'interdiction de tous les états du système associés au concept de super-état. Par exemple, l'interdiction du super-état *TV\_Allumée* permet d'interdire l'allumage de toutes les TV du système.

La gestion des modes, issue du 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, Hamani *et al.* 2004], 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.

### 2.3.2.3 Activation et définition

Pour compléter la définition des interactions, il est indispensable d'introduire les conditions d'activation et de préciser leur domaine de définition. Ainsi, une interaction se définit par le couple  $\{C, A\}$  où  $C$  représente le type d'activation et  $A$  le type d'action (*niveau sémantique* ou *niveau de restriction*). Nous définissons pour cela trois types d'activation à partir desquelles les services et les modes peuvent être activés :

- $C1$  : sous la demande de l'utilisateur
- $C2$  : automatiquement avec confirmation de l'utilisateur
- $C3$  : automatiquement sous une condition prédéfinie

Le domaine de définition des interactions est présenté dans la figure 2.13. Les services et les modes sont définis respectivement par leur niveau sémantique et leur niveau de restriction (suivant l'axe horizontal de la figure). Le domaine de définition de ces interactions varie suivant le type d'interaction : le troisième niveau sémantique et le troisième niveau de restriction sont définis au niveau *système*, alors que les deux autres le sont au niveau *ressource* (composant élémentaire ou agrégé). Un service élémentaire peut par exemple être associé à un composant élémentaire mais aussi à un composant agrégé alors que les modes sont obligatoirement associés aux composants agrégés.

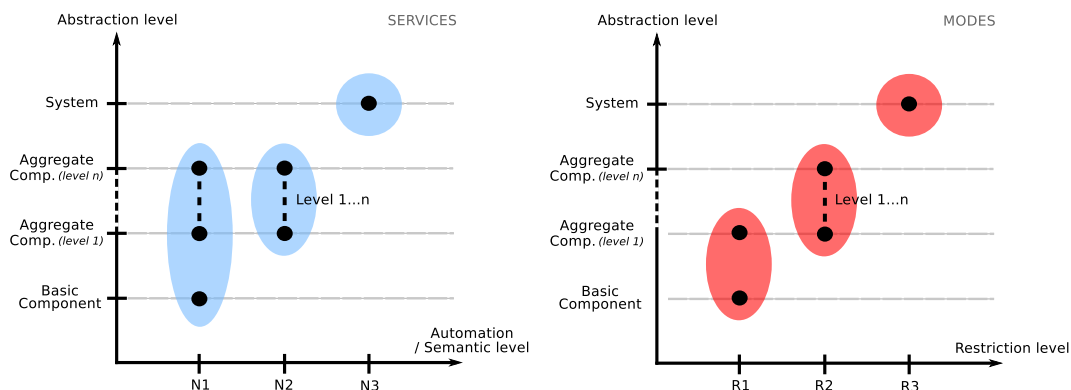


FIGURE 2.13 – Taxonomie d'interactions (services et modes) et domaine de définition

En plus de l'environnement domotique modélisé par agrégation de composants, il est donc possible de définir des interactions adaptées aux besoins de l'utilisateur : différents types de services et de modes peuvent ainsi être définis et soumis à différents types d'activation. Une évaluation auprès de personnes non-expertes en domotique (étudiants et ergothérapeutes), présentée dans le chapitre 5, permet de mettre en avant l'expressivité et l'accessibilité du DSML.



## **2.4 Conclusion du chapitre / contribution**

Dans ce chapitre, un langage de modélisation dédié (DSML) au domaine de la domotique d'assistance a été présenté. Ce DSML, qui se veut accessible à des personnes non-expertes en domotique, est composé de deux vues : une vue physique pour représenter l'environnement domotique modélisé par agrégation de composants et une vue fonctionnelle pour définir les interactions entre l'utilisateur et son système domotique. Dans cette dernière vue, deux types d'interactions sont définis pour prendre en compte un nombre de besoins utilisateur le plus grand possible. Ainsi, il est possible de définir des requêtes de différents types (i.e. services) comme des interdictions temporaires (i.e. modes) sur des ressources domotiques. Le chapitre suivant présente le flot de conception dans lequel s'inscrit ce langage de modélisation, qui permet d'aboutir à l'implantation physique du système domotique modélisé.

# 3

## Flot de conception dirigé par les modèles

*“Besides black art, there is only automation and mechanization.”*

Federico García Lorca, 1936

Le concepteur non-expert en domotique, que nous cherchons à intégrer dans le processus de conception d'un système domotique, doit intervenir à un haut niveau d'abstraction sans se préoccuper des problèmes d'implémentation et d'intégration des différentes technologies domotiques. Pour adapter le système domotique aux besoins des utilisateurs, ce dernier a besoin de méthodes et d'outils adaptés à ses compétences techniques. L'IDM, qui pose comme principe que tout ou partie d'une application informatique est généré à partir de modèles, nous semble pertinente pour répondre à ces besoins. L'automatisation du processus de conception domotique est rendue possible par des transformations de modèles, faisant ainsi évoluer les modèles décrivant le système considéré de l'état passif (i.e. système modélisé à un haut niveau d'abstraction) à l'état actif (i.e. implémentation logicielle). L'objectif de ce chapitre est de présenter le flot de conception proposé dans ces travaux. En utilisant l'IDM, nous proposons de mettre en oeuvre un processus de conception intégrant l'utilisateur dépendant et le concepteur non-expert en domotique (section 3.1). À partir de modèles indépendants de toute plateforme définis à l'aide du DSML présenté dans le chapitre précédent, la configuration propre à une plateforme domotique peut être générée en suivant le flot de conception proposé (section 3.2). Une implémentation possible de ce flot est proposée en intégrant la plateforme Danah, développée au laboratoire Lab-STICC. Finalement, un logiciel de configuration composé d'un éditeur graphique est développé pour outiller et mettre en application le flot (section 3.3).

## Sommaire

---

|         |  |    |
|---------|--|----|
| 3.1     | Introduction au flot de conception . . . . .         | 75 |
| 3.1.1   | Processus de conception centré utilisateur . . . . . | 75 |
| 3.1.1.1 | Caractéristiques des acteurs . . . . .               | 75 |
| 3.1.1.2 | Intégration des acteurs dans le processus . . . . .  | 76 |
| 3.1.2   | Flot de conception théorique . . . . .               | 77 |
| 3.1.2.1 | Architecture dirigée par les modèles . . . . .       | 77 |
| 3.1.2.2 | Flot de conception proposé . . . . .                 | 78 |
| 3.2     | Implémentation du flot de conception . . . . .       | 80 |
| 3.2.1   | Intégration du PSM Danah . . . . .                   | 80 |
| 3.2.1.1 | Présentation de Danah . . . . .                      | 80 |
| 3.2.1.2 | Méta-modèle Danah . . . . .                          | 80 |
| 3.2.2   | Flot de conception pratique . . . . .                | 82 |
| 3.2.2.1 | Résumé des étapes de conception . . . . .            | 82 |
| 3.2.2.2 | Déroulement des étapes du flot . . . . .             | 82 |
| 3.3     | Outillage du flot de conception . . . . .            | 90 |
| 3.3.1   | IntelHome Designer . . . . .                         | 90 |
| 3.3.1.1 | Organisation du logiciel . . . . .                   | 90 |
| 3.3.1.2 | Tableau de bord . . . . .                            | 92 |
| 3.3.2   | Éditeur graphique . . . . .                          | 93 |
| 3.3.2.1 | Modèle d'environnement . . . . .                     | 93 |
| 3.3.2.2 | Modèle d'interactions . . . . .                      | 94 |
| 3.4     | Conclusion du chapitre / contribution . . . . .      | 97 |

---

## 3.1 Introduction au flot de conception

Deux acteurs peuvent être identifiés dans nos travaux. Il s'agit du concepteur et de l'utilisateur. Dans le cadre de la domotique d'assistance, ces deux acteurs présentent des profils originaux par rapport au contexte standard d'utilisation de la domotique (section 3.1.1). Dans ces travaux, nous proposons d'intégrer ces derniers dans le flot de conception dirigé par les modèles présenté dans la section 3.1.2.

### 3.1.1 Processus de conception centré utilisateur

#### 3.1.1.1 Caractéristiques des acteurs

Le langage de modélisation et le flot de conception associé doit répondre aux besoins des deux acteurs identifiés dans nos travaux. Il s'agit du *concepteur* et de l'*utilisateur*. Comme cela a déjà été présenté dans la section 1.3.2.1 du chapitre introductif, ces acteurs présentent des profils originaux par rapport au contexte standard d'utilisation de la domotique. Nous précisons dans cette section leurs caractéristiques particulières.

**L'utilisateur :** peut présenter des capacités motrices réduites et/ou des troubles cognitifs, à des stades plus ou moins avancés. À la pluralité des situations de handicap, le langage de conception domotique proposé dans ces travaux doit être suffisamment expressif pour répondre à un nombre de besoins satisfaisant. En outre, des interactions de haut niveau sémantique sont essentielles pour les raisons suivantes :

- Le contrôle efficace de plusieurs équipements simultanément devient parfois difficile à assurer pour l'utilisateur. En proposant des interactions intuitives, la charge cognitive<sup>1</sup> 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" regroupées sous une unique action.

**Le concepteur :** 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. Pour cette per-

---

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].

sonne, un système domotique d'assistance doit avant tout être facile à concevoir, mais aussi facile à maintenir, facile à commander et flexible. Elle 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.

### 3.1.1.2 Intégration des acteurs dans le processus

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 (pour que la conception domotique puisse être accessible au concepteur non-expert) ou de l'utilisation du système domotique (pour améliorer la situation de handicap de l'utilisateur). Pour cela, nous proposons une méthodologie intuitive basée sur un processus de conception centré utilisateur [Maguire 2001] (*Human Centered Design*, ISO 13407), appliqué à la domotique. Ce processus a été présenté succinctement sur la figure 1.8 du chapitre introductif. Nous en précisons ici les étapes qui s'appliquent aux acteurs identifiés :

1. **Analyse** préliminaire où le concepteur non-expert en domotique précise les attentes et les besoins de l'utilisateur en terme d'adaptations domotiques.
2. **Conception** séparée en spécification du contexte d'utilisation et spécification des besoins associés. Cette étape est assurée par le concepteur non-expert qui utilise le langage de modélisation proposé pour définir l'environnement domotique, puis l'adapter en définissant des interactions (services et modes).
3. **Production** des solutions à partir des modèles définis dans l'étape précédente. Cette étape consiste à générer automatiquement la commande du système domotique modélisé par transformations successives de modèles.
4. **Évaluation** de la solution produite. Le concepteur non-expert et l'utilisateur évaluent conjointement la configuration générée pour l'application domotique en situation réelle. Dans le cas où le système ne convient pas ou plus à l'utilisateur (évolution des besoins), le processus de conception est réitéré.

Pour la mise en oeuvre informatique d'un tel processus, nous utilisons l'IDM. Des modèles indépendants de toute plateforme sont définis par le concepteur non-expert à l'aide du langage de modélisation présenté dans le chapitre précédent. Cela lui permet d'assurer les étapes de spécification du contexte d'utilisation (modèle d'environnement) et de spécification des besoins de l'utilisateur (modèle d'interactions) indépendamment de la plateforme technologique utilisée. Les processus de transformation de modèles permettent alors de garantir la transition entre les différentes étapes de la méthodologie et de produire les solutions / implémentations logicielles conformément aux spécifications préalablement définies.

### 3.1.2 Flot de conception théorique

#### 3.1.2.1 Architecture dirigée par les modèles

Nous pensons que l'utilisation conjointe d'un DSML 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 de modélisation dédié permet i) de répondre à des besoins particuliers en ne conservant que les concepts métier qui font sens dans ce domaine et ii) de s'abstraire des spécificités matérielles pour ne s'intéresser qu'aux spécifications fonctionnelles. Associé à une architecture dirigée par les modèles, il permet aux modèles définis à l'aide de ce langage de passer de l'état passif (i.e. visualisation) à l'état actif pour l'implémentation sur une plateforme cible.

La spécification MDA (*Model Driven Architecture*) [Miller *et al.* 2001] propose de séparer explicitement les spécifications fonctionnelles de l'architecture d'exécution. En élevant ainsi le niveau d'abstraction, le concepteur non-expert en domotique ne se préoccupe pas de l'implémentation du système, automatisée à travers l'utilisation de transformations de modèles. La génération automatique de la 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 du DSML vers un modèle spécifique à une plateforme particulière, pour finalement générer le code implémentable. L'architecture MDA offre ainsi une solution flexible et évolutive quant à l'intégration de nouvelles technologies domotiques : seule la partie spécifique à une plateforme est amenée à changer.

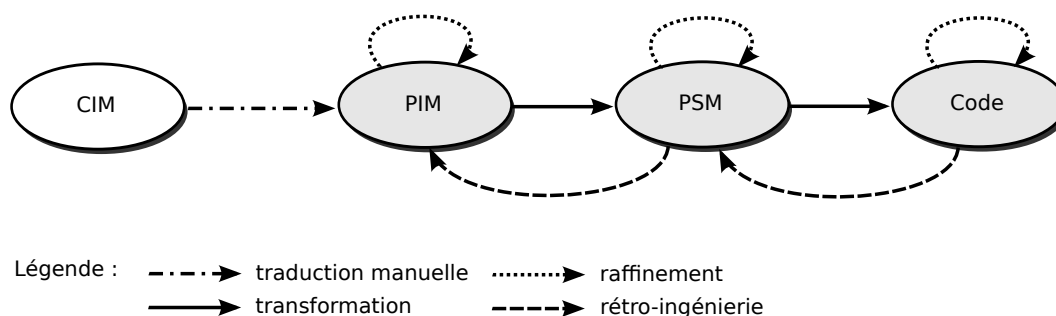


FIGURE 3.1 – Niveaux de modélisation de la spécification MDA

La figure 3.1 présente les niveaux de modélisation de la spécification MDA :

- Le niveau **CIM** (*Computation Independent Model*) représente le niveau le plus abstrait de représentation sans moyen informatique. Dans notre contexte, il représente l'étape d'analyse préliminaire du processus de conception centré utilisateur.

Exempt de toute informatisation, le passage au niveau de modélisation suivant s'effectue généralement par traduction manuelle ou orale.

- Le niveau **PIM** (*Platform Independent Model*) représente le premier niveau de modélisation informatique qui se veut indépendant de toute plateforme. Le concepteur manipule à ce niveau le DSML pour définir l'environnement domotique et les interactions de l'utilisateur avec ce dernier, sans utiliser de termes techniques liés à l'implémentation du système domotique. Le passage au niveau de modélisation suivant passe par une ou plusieurs transformations de modèles.
- Le niveau **PSM** (*Platform Specific Model*) contient toutes les informations techniques relatives à la/aux plateforme(s) domotique(s) ciblée(s). Ce modèle ne pouvant pas être exécuté, il nécessite une dernière transformation de type "modèle → texte" (*Model To Text*) pour garantir l'implémentation logicielle.
- Le niveau **Code** comprend les artefacts exécutables et conformes aux modélisations issues des étapes précédentes. Dans notre contexte, il s'agit d'un fichier structuré indispensable pour la configuration de l'application domotique.

### 3.1.2.2 Flot de conception proposé

Le flot de conception dirigé par les modèles proposé dans ces travaux permet d'automatiser le processus de conception de systèmes domotiques (cf figure 3.2). Il comprend 5 grandes étapes (manuelles en orange, automatiques en vert) qui permettent d'aboutir à la configuration domotique à partir de modèles définis par le concepteur non-expert.

1. **Modélisation de l'Environnement** : après avoir sélectionné les fichiers de configuration (e.g. *ETS, LIRC*) pour récupérer automatiquement les commandes (e.g. *KNX/EIB, infrarouge*) des équipements présents dans l'habitat, le concepteur non-expert commence la première activité de modélisation. À partir d'une bibliothèque, il peut intégrer et adapter des composants (e.g. équipements et pièces) pour modéliser l'environnement domotique (cf approche ascendante présenté dans la section 2.3.1.2 du chapitre 2).
2. **Transformation Environnement → Interactions** : cette transformation de modèles permet de passer d'une représentation physique considérant l'environnement domotique à une représentation logique du système considérant les interactions entre l'utilisateur et ce dernier. Chaque opération associée aux composants (élémentaires ou agrégés) du modèle d'environnement est ainsi transformée en service élémentaire activable manuellement. Par ailleurs, une liste d'états est générée pour permettre leur interdiction à travers les modes.
3. **Modélisation des Interactions** : à la fin de l'étape précédente, on aboutit à un premier modèle d'interactions qu'il est possible d'enrichir par la suite pour répondre aux besoins de l'utilisateur. Le concepteur peut soumettre certains services à des conditions d'activation automatiques, définir des scénarios avec ou non prise en

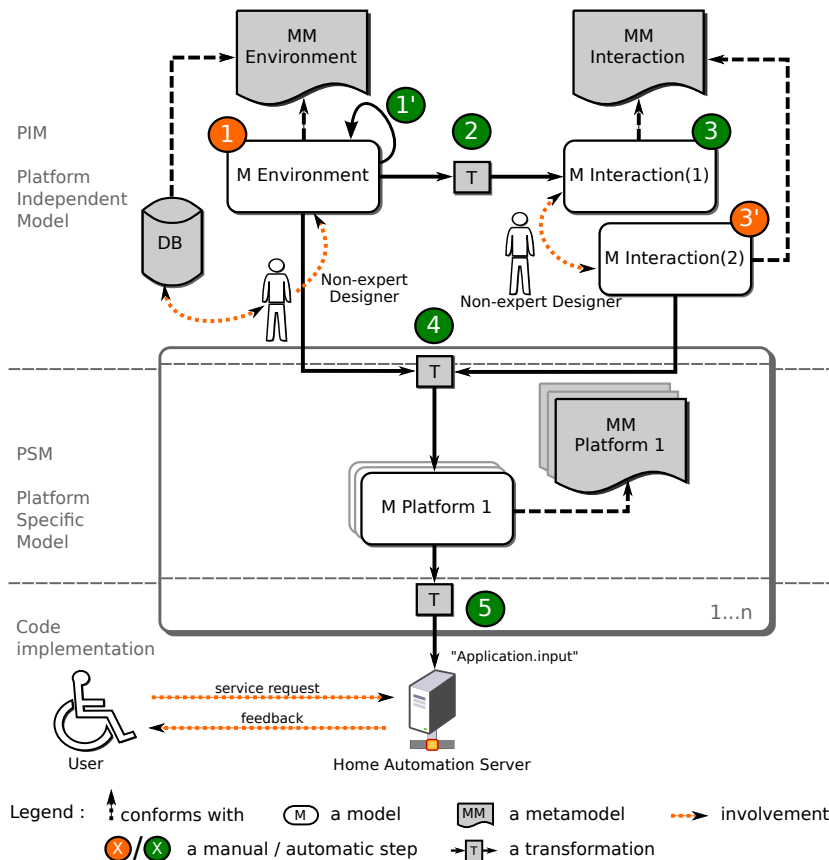


FIGURE 3.2 – Flot de conception théorique basé sur une architecture MDA

compte d'aléas de fonctionnement, définir des modes d'interdiction pour restreindre temporairement l'accès à certains services, etc.

4. **Transformation PIM → PSM** : une fois les modèles indépendants de toute plateforme définis - modèle d'environnement (étape 1) et modèle d'interactions (étape 2) - cette transformation permet de passer d'une description fonctionnelle à un modèle spécifique à une plateforme. À partir des commandes associées à chaque opération, il est possible de générer le fichier de configuration d'une plateforme domotique conformément à son méta-modèle.
5. **Transformation PSM → Code** : à partir du modèle spécifique à la plateforme domotique, le code respectant sa syntaxe est automatiquement généré par une transformation de type "modèle → texte". Le fichier de configuration en résultant peut finalement être intégré dans un serveur domotique à partir duquel l'utilisateur peut solliciter les services et modes préalablement définis.



Le flot de conception offre une certaine flexibilité et évolutivité quant à l'intégration d'une nouvelle plateforme technologique : seule la partie spécifique (i.e. modèle PSM et transformations associées) doit être définie pour adresser une nouvelle plateforme domotique. Une implémentation possible de ce flot pour la plateforme Danah est proposée dans la section suivante : toutes les étapes du flot y sont reprises et illustrées sur des exemples simples.

## 3.2 Implémentation du flot de conception

L'implémentation du flot de conception proposé dans ces travaux intègre la plateforme Danah (section 3.2.1). Il en résulte un flot de conception pratique (section 3.2.2) permettant de générer la configuration domotique de cette plateforme à partir du langage de modélisation indépendant de toute plateforme.

### 3.2.1 Intégration du PSM Danah

#### 3.2.1.1 Présentation de Danah

Danah [Lankri *et al.* 2008] est la plateforme domotique cible retenue dans nos travaux. Il s'agit d'un middleware<sup>2</sup> pour les 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 en utilisant des bibliothèques libres et open-sources pour la prise en charge de différents protocoles de communication (infrarouge via *LIRC*, *KNX/EIB* via *eibd* [Kastner *et al.* 2005], bluetooth via la bibliothèque *BlueZ*). Depuis [Lankri *et al.* 2008], nous avons développé ce middleware pour prendre en compte les concepts introduits dans ces travaux (i.e. intégration des modes, des activations semi-automatiques, etc.).

Le middleware Danah est basé sur une architecture client-serveur (cf figure 3.3). Les bornes domotiques (serveurs Danah), matérialisées par des cartes *IGEPv2* (basées sur des processeurs *ARM / OMAP3530* [ISEE 2010]), sont distribuées dans l'environnement. Une borne peut ainsi être connectée à un bus domotique *KNX-EIB* pour contrôler les équipements qui y sont connectés. D'autres bornes sont chargées de contrôler les équipements à activation locale via infrarouge ou bluetooth comme les équipements multimédia. L'utilisateur, équipé d'un *PDA (Personal Digital Assistant)* embarqué dans son fauteuil, a la possibilité de solliciter les services et les modes publiés par ces bornes sur son IHM.

---

2. Un middleware domotique permet de faire abstraction des couches basses de communication pour se focaliser sur la notion de services proposés à la personne.

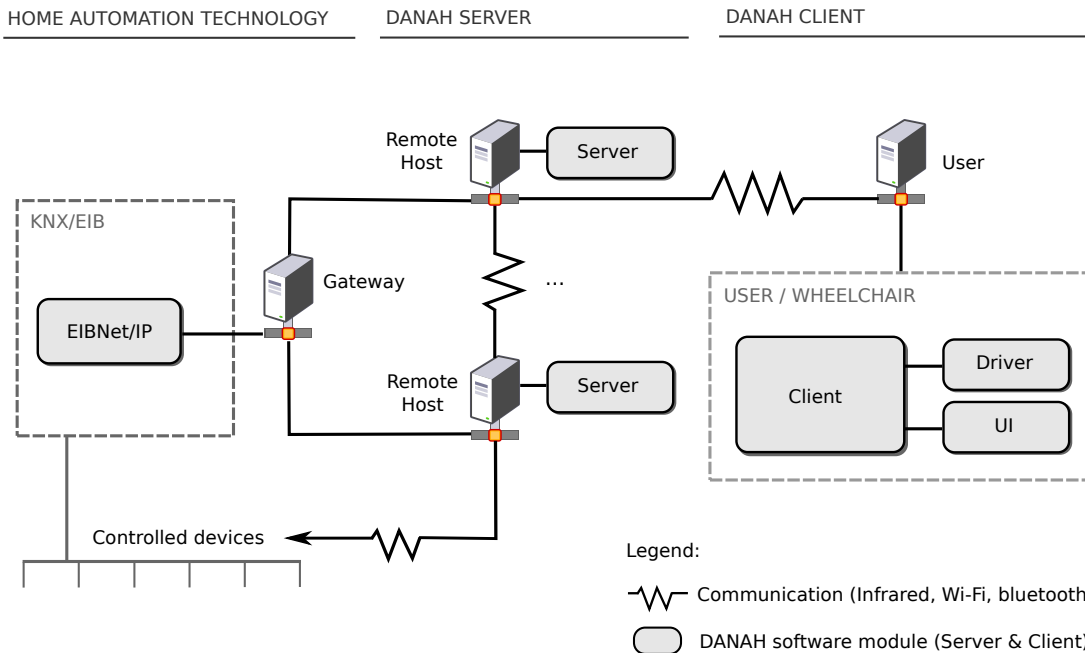
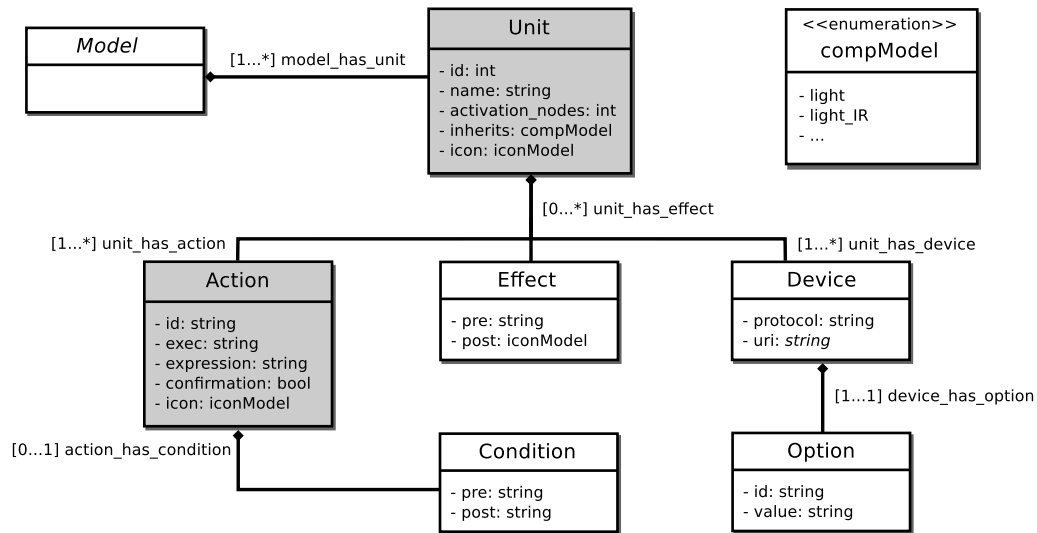


FIGURE 3.3 – Architecture de déploiement du middleware Danah

### 3.2.1.2 Méta-modèle Danah

Pour intégrer la plateforme Danah dans le flot de conception, nous avons défini un méta-modèle décrivant la structure de ce middleware. À partir de ce dernier, il est possible de générer un fichier de configuration à intégrer dans les bornes domotiques en reprenant les services et modes définis dans les étapes de modélisation.

La figure 3.4 présente un extrait du méta-modèle Danah. Ce dernier est constitué principalement de deux niveaux : **Unit** et **Action** (cf classes grisées), que l'on retrouve sur l'IHM de l'application *Danah\_Client*. Les autres éléments du méta-modèle Danah sont liés au protocole de communication et au fonctionnement du système domotique, cachés à l'utilisateur. Les *Units* représentent les éléments du système domotique que l'utilisateur peut commander. Elles sont composées entre autres d'un module *device* permettant d'en renseigner les spécificités matérielles. Les attributs *protocol* et *uri* décrivent respectivement le protocole à charger pour ce composant et l'emplacement de sa librairie. Le module *option* contient les propriétés *id* et *value* permettant de renseigner l'identifiant, l'adresse du composant sur le bus ou sur un réseau (suivant le protocole utilisé). Les *Actions* représentent les commandes domotiques que l'utilisateur peut solliciter. Ces commandes peuvent être décrites de deux manières. La première consiste à définir une commande directement associée au protocole de communication d'un équipement (e.g. "SEND\_ONCE TV1 CH1" permet de lancer une commande - la chaîne 1 de la télévision TV1 - listée dans le fichier de configuration "lircd.conf"). La deu-

FIGURE 3.4 – Extrait du méta-modèle *Danah*

xième manière consiste à définir une expression pour les scénarios avec prise en compte d'aléas (e.g. "SEQOR(L1.SwitchON L2.SwitchON)" est un scénario respectant la syntaxe Danah [Lankri *et al.* 2008] qui allume la lampe L2 dans le cas où la lampe L1 n'a pas été correctement allumée.).

## 3.2.2 Flot de conception pratique

### 3.2.2.1 Résumé des étapes de conception

Le flot de conception dirigé par les modèles permet au concepteur non-expert en domotique de manipuler des modèles indépendants de toute plateforme. 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) pour définir les interactions permettant de répondre aux besoins de l'utilisateur (étape 3). Ces deux modèles, décrits à l'aide du DSML, sont ensuite transformés en un modèle spécifique à une plateforme domotique.

Le flot de conception théorique présenté sur la figure 3.2 a été repris et implémenté pour la plateforme domotique Danah. Les deux modèles indépendants de toute plateforme sont transformés en un modèle spécifique à la plateforme Danah (étape 4) pour finalement générer un fichier de configuration à intégrer dans une borne domotique (étape 5). Ce fichier reprend les services et les modes définis dans l'étape de modélisation indépendante de toute plateforme.

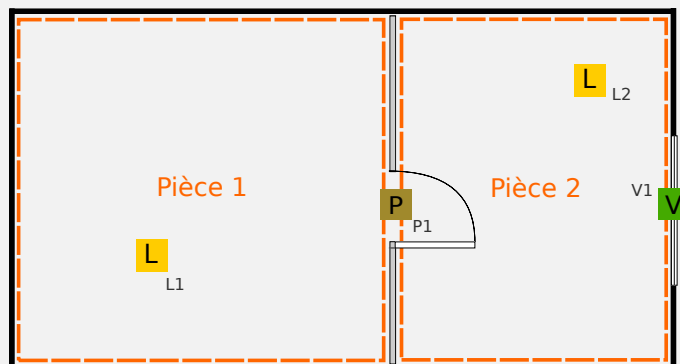
### 3.2.2.2 Déroutement des étapes du flot

Chaque étape de ce flot est commentée et illustrée en prenant un cas d'étude simple pour vérifier l'applicabilité de la méthodologie de conception retenue. L'illustration proposée ne considère pas la syntaxe concrète (syntaxe graphique qui sera introduite dans la section 3.3.2) mais considère uniquement la syntaxe abstraite présentée dans le chapitre 2, indépendante de l'outil informatique utilisé<sup>3</sup>. Les étapes du flot présentées ci-dessous font toutes références au flot de conception présenté dans la figure 3.2.

**Étape 1 : Modélisation de l'environnement** En respectant l'approche ascendante de conception préconisée pour la première étape du flot (des composants élémentaires jusqu'au système complet), le premier travail du concepteur est de choisir les composants élémentaires (i.e. équipements domotiques). Le concepteur choisit ensuite d'assembler ces composants pour former des composants de plus haut niveau (e.g. à l'échelle d'une pièce, puis de l'habitat complet). Pour intégrer les portes dans la structure hiérarchique représentant l'habitat, le concepteur renseigne un lien topologique (avec l'association *door\_link*) entre les composants définis à l'échelle d'une pièce et cette dernière. Pour contrôler les équipements domotiques, le concepteur a finalement besoin d'intégrer un ou plusieurs serveurs domotique(s) et/ou un bus domotique.

#### CAS D'ÉTUDE

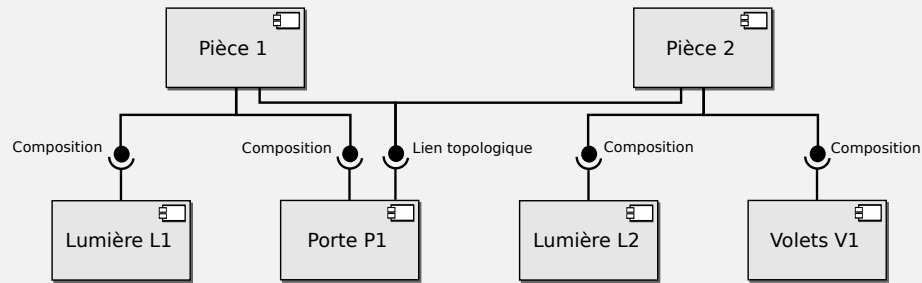
Considérons deux pièces "Pièce 1" et "Pièce 2" dans lesquelles se trouvent les équipements suivants : les lumières  $L_1$  et  $L_2$ , la porte  $P_1$  et les volets  $V_1$ . La porte  $P_1$  sépare ces deux pièces :



À partir des équipements  $L_1$ ,  $P_1$ ,  $L_2$  et  $V_1$  de notre cas d'étude, les composants agrégés *Pièce 1* et *Pièce 2* sont définis. Dans ce cas d'étude, la porte  $P_1$

3. Pour une illustration complète de la méthodologie avec la syntaxe graphique et le logiciel développé sur un cas d'étude réel, se reporter au chapitre 5, section 5.1.3.

fait partie de la *Pièce 1* (lien de composition) et est associée aux composants représentant les deux pièces par l'intermédiaire d'un lien topologique (Il s'agit d'un choix arbitraire qui n'a aucune incidence sur la modélisation du système : la porte *P1* aurait pu faire partie de la *Pièce 2*). Il en résulte une structure hiérarchique de six composants, tous liés par des relations de composition ou d'association :



Une fois l'environnement modélisé de manière conforme à son méta-modèle (cf section 2.3.1 du chapitre 2), une première transformation est lancée (étape 1' du flot). Il s'agit d'une transformation endogène (i.e. qui s'applique sur le même méta-modèle) qui génère les opérations des composants agrégés par rapport aux opérations des composants de niveau hiérarchique inférieur. L'algorithme 1 illustre la représentation algorithmique de la règle "*operation2operation*". Considérons un composant de niveau  $L + 1$  :  $c_{L+1}$ . En appliquant cette règle de transformation, le composant  $c_{L+1}$  se voit attribuer les opérations  $o_{c_L}$  du composant de niveau inférieur  $c_L$  qu'il contient.

---

**Algorithm 1** Représentation algorithmique de la règle *operation2operation*

---

```

for all component  $c_{L+1}$  do
  for all component  $c_L \in c_{L+1}$  do
    for all operation  $o_{c_L}$  do
       $o_{c_{L+1}} \leftarrow o_{c_L}$ 
    end for
  end for
end for
  
```

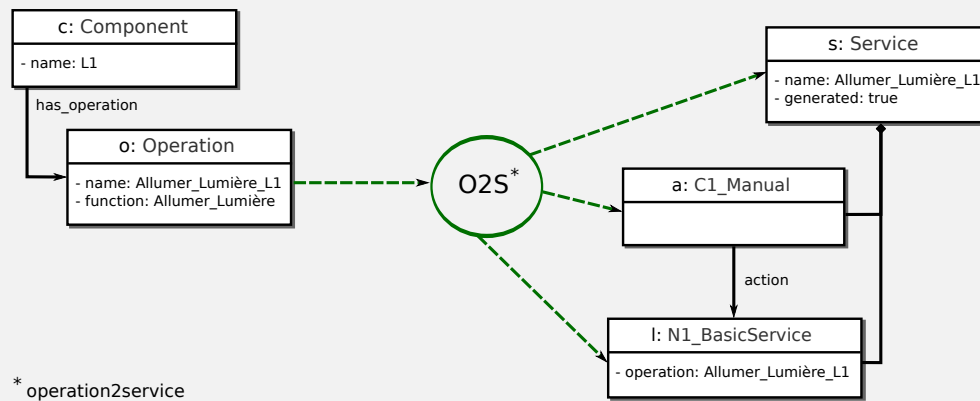
---

Cette transformation de modèles s'applique aux états des ressources de la même manière. Les états et opérations des ressources de niveau  $L$  (en partant des composants élémentaires) sont donc tous deux transmis par héritage aux composants de niveau  $L + 1$ , et ce de manière récursive jusqu'à aboutir aux composants représentant le système complet (les fonctions et super-états sont définis à ce niveau.).

**Étape 2 : Transformation environnement → interactions** 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 (étape 2 du flot). Ce changement de vue se traduit par l'utilisation de la transformation de modèles "Env2Int". La principale règle de transformation concerne la transformation d'opérations en services élémentaires activables manuellement : activation de type *c1*, niveau sémantique *n1* (cf section 2.3.2 du chapitre 2). L'algorithme 2 illustre la représentation algorithmique de cette règle ("operation2service").

**CAS D'ÉTUDE**

Reprenons la lumière *L1* et ses opérations *Allumer\_Lumière* et *Éteindre\_Lumière*. En appliquant la règle de transformation "operation2service", ces deux opérations sont transformées en deux services *Allumer\_Lumière\_L1* et *Éteindre\_Lumière\_L1* activables manuellement. La figure ci-dessous montre l'application de cette règle sur l'opération *Allumer\_Lumière\_L1*.



En appliquant la règle de transformation "operation2service" (cf algorithme 2) sur l'opération (*o : Allumer\_Lumière\_L1*), un service *s* de niveau sémantique élémentaire (*l : N1\_Basic\_Service*) activable manuellement (*c : C1\_Manual*) est généré (cf section 2.3.2 du chapitre 2).

Les états associés à chaque opération sont générés de la même manière. Cela permet d'aboutir à un modèle d'interactions minimaliste (cf section 2.3.2 du chapitre 2), comprenant un certain nombre d'états (qu'il sera par la suite possible d'interdire) et autant

**Algorithm 2** Représentation algorithmique de la règle *operation2service*


---

```

for all basic_component c do
  for all operation o do
    new service s
    s.name ← o.name
    s.condition ← c1
    s.semanticslevel ← n1
  end for
end for

```

---

de services élémentaires que le concepteur pourra enrichir.

**Étape 3 : Modélisation des interactions** Cette étape propose d’enrichir le modèle d’interactions automatiquement généré pour l’adapter aux besoins de l’utilisateur (passage de l’étape 3 à l’étape 3’ du flot). À partir des services élémentaires présents, il est possible de créer des scénarios et de les soumettre à des conditions d’activation automatique ou avec demande de confirmation. De la même manière, les modes d’interdiction peuvent être définis et soumis à des conditions particulières.

**CAS D’ÉTUDE**

Dans un premier temps, l’utilisateur de notre cas d’étude a besoin d’automatiser le fonctionnement de l’habitat. Pour cela, le concepteur définit des scénarios de vie (enchaînement de plusieurs services à partir d’une seule activation) avec des conditions d’activation automatique (cf service 3.1 ci-dessous) ou avec demande de confirmation de l’utilisateur avant activation (cf service 3.2). La représentation textuelle des interactions présentées dans ce cas d’étude reprend les éléments du méta-modèle d’interactions défini dans le chapitre précédent. L’activation automatique est par exemple représentée par le terme “*SystExecute*)” qui fait référence au concept *C3\_Automatic* du méta-modèle. Le terme “*SystDemande*” fait référence quant à lui au concept *C2\_SemiAutomatic*.

*Soir1* :  
 $Heure = 22h \Rightarrow SystExecute(AllumerLumiere\_L1 \text{ et } DescendreVolet\_V1)$  (3.1)

*Soir2* :  
 $V1.Descendus \Rightarrow SystDemande(AllumerLumiere\_L1 \text{ et } DescendreVolet\_V1)$   
 (3.2)

Pour prévenir des éventuelles pannes des actionneurs, il est possible de définir des scénarios avec prise en compte du résultat d’exécution. Le service 3.3 est activable manuellement (le terme “*UtilisateurVeut*” correspond au concept *C1\_Manual* du méta-modèle d’interactions) et permet d’ouvrir la porte et

d'allumer la lumière  $L2$  dans le cas où la lumière  $L1$  ne s'est pas correctement activée (symbole  $\otimes$ ).

*Allumer :*

$UtilisateurVeut(AllumerLumiere\_L1 \otimes (OuvrirPorte\_P1 \text{ et } AllumerLumiere\_L2))$   
(3.3)

De manière complémentaire aux services, il est possible de définir des restrictions sur l'environnement domotique. La définition de modes d'interdiction permet ainsi de restreindre momentanément l'accès aux services domotiques depuis l'interface de l'utilisateur. Pour illustrer ce type d'interactions, le mode 3.4 interdit de fermer la porte  $P1$  entre 22h et 8h.

$AntiFugue : 22h < Heure < 8h \Rightarrow SystInterdit(P1.Ouverte)$  (3.4)

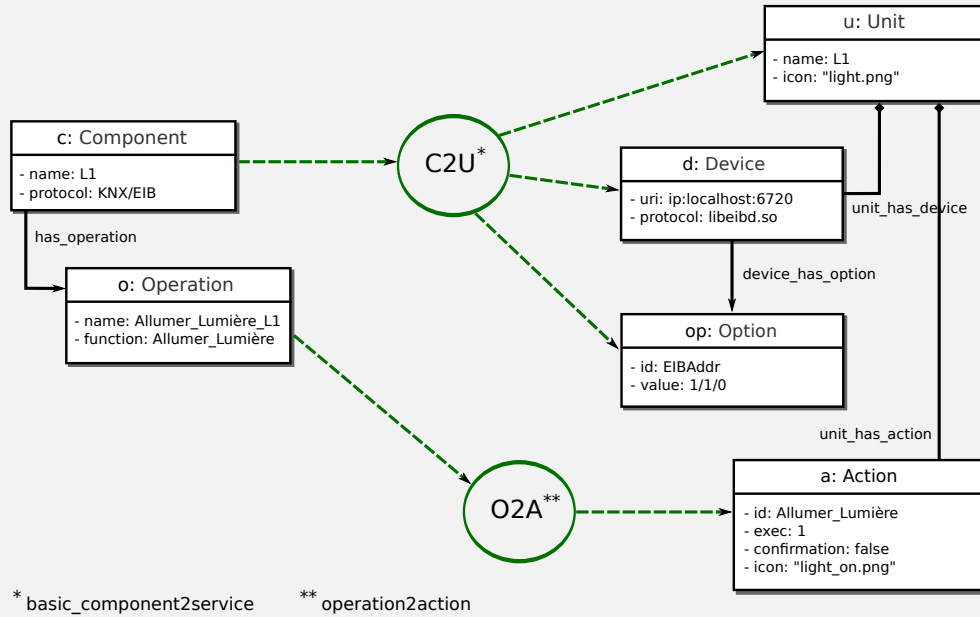
**Étape 4 : Transformation  $PIM \rightarrow PSM$**  À partir des modèles indépendants de toute plateforme définis dans les étapes 1 et 3 du flot, il est maintenant possible de générer un modèle spécifique à la plateforme domotique Danah. Cette étape (étape 4 du flot) consiste donc à reprendre les éléments définis dans le modèle d'environnement et le modèle d'interactions et à les rendre conformes au méta-modèle Danah. Ce dernier est composé de deux niveaux : i) *Unit* qui représente les équipements qui peuvent être contrôlés et ii) *Action* qui représente les commandes que l'utilisateur peut demander. La transformation "*Intel2Danah*" permet de générer ces deux niveaux à partir des deux modèles définis à l'aide du *DSML*. Les *Units* sont générées à partir du modèle d'environnement. Il en est de même pour les informations *protocol*, *uri*, *id*, *value* relatives au protocole de communication utilisé pour chaque composant. Les *Actions* sont générées à partir du modèle d'interactions. Les paramètres *expression* ou *exec* y sont définis suivant le type de services considéré. Le paramètre *expression* est défini lorsque l'interaction considérée est un scénario ou un mode d'interdiction (e.g. l'expression "*SEQAND(L1.SwitchON L2.SwitchON)*" permet de décrire un scénario comprenant l'activation de la lumière  $L1$  puis  $L2$  dans Danah.). Le paramètre *exec* est défini lorsqu'il s'agit d'un service associé à un composant élémentaire (e.g. l'exécution "*SEND\_ONCE TV1 CH1*" permet de lancer la commande infrarouge *TV1.CH1* qui se trouve dans le fichier de configuration "*lircd.conf*").

## CAS D'ÉTUDE

Reprenons l'exemple de la lampe  $L1$ , définie comme un composant utilisant le protocole *KNX/EIB*. La transformation "*Intel2Danah*" permet entre autres de générer une *unit* et deux *actions* pour ce composant. La figure ci-dessous montre l'application de deux règles de transformation ("*ba-*



*sic\_component2service*” et *“operation2action”*) sur *L1* et une de ces opérations (*Allumer\_Lumière\_L1*).



En appliquant la règle de transformation *“basic\_component2unit”* sur le composant *L1* (*c : Component*), une *unit* est créée (*u : Unit*). Le protocole *KNX/EIB* renseigné dans l’étape de modélisation de l’environnement est utilisé pour générer les éléments spécifiques à la plateforme *KNX* (*d : Device* et *op : Option*) et leurs paramètres. La règle *“operation2action”* génère une *action* (*a : Action*) à partir de l’opération *“Allumer\_Lumière\_L1”* (*o : Operation*).

À ce niveau, le concepteur non-expert doit effectuer une correspondance entre les opérations de chaque composant défini dans le modèle d’environnement, et une liste de commandes extraites de fichiers de configuration importés au début du flot de conception. Pour des équipements *KNX/EIB*, il s’agit d’un fichier exporté depuis le logiciel *ETS* [KNX Association 1993]. Sur la figure du cas d’étude ci-dessus, le paramètre *value* de l’élément *Option* qui fait référence à l’adresse de l’équipement sur le bus domotique, a été extrait à partir d’un fichier de configuration *ETS*.

**Étape 5 : Transformation *PSM* → *Code*** À partir du *PSM* issu de l’étape précédente, il est possible de générer un fichier de configuration pour la plateforme domotique Danah. Un générateur de code traduit chaque élément du *PSM* en texte. Cela permet d’aboutir à un fichier de configuration structuré respectant la syntaxe Danah.

## CAS D'ÉTUDE

Le PSM Danah reprend les équipements domotiques (*L1*, *P1*, *L2* et *V1*), les pièces (*Pièce 1* et *Pièce 2*) et interactions domotiques (services *Soir1*, *Soir2*, *Allumer* et mode *AntiFermeture*) précédemment définis. À partir de ce modèle, il est possible de générer le fichier de configuration pour une borne domotique Danah. La figure ci-dessous montre un extrait du code généré pour la lumière L1 de notre cas d'étude.

```
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"
    }
  }
}
```

Une fois le fichier de configuration généré, ce dernier est embarqué dans une borne

domotique supportant la plateforme Danah. L'utilisateur peut ainsi s'y connecter pour solliciter les interactions précédemment définies. La figure 3.5 donne un aperçu de l'interface graphique présente sur le PDA de l'utilisateur.

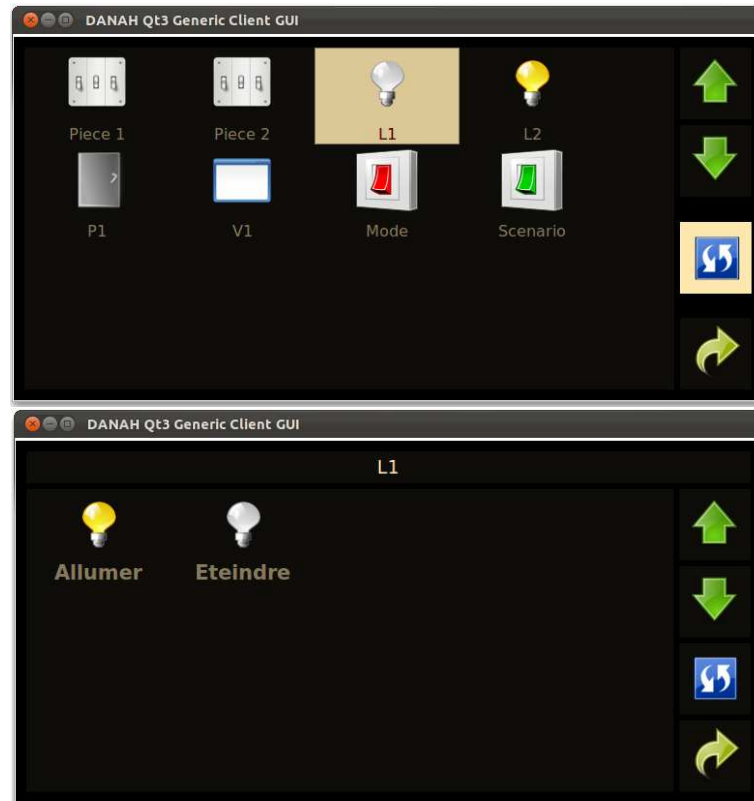


FIGURE 3.5 – Interface utilisateur Danah affichée sur le PDA

### 3.3 Outillage du flot de conception

Pour outiller le flot de conception, un logiciel baptisé *IntelHome Designer* est développé. Basé sur des outils de modélisation existants, il permet de mettre en oeuvre les différentes étapes du flot (section 3.3.1). Pour permettre la manipulation du DSML par une personne non-experte, un éditeur graphique est défini (section 3.3.2).

#### 3.3.1 IntelHome Designer

##### 3.3.1.1 Organisation du logiciel

*IntelHome Designer* vise à fournir au concepteur non-expert un cadre logiciel lui permettant de participer aux étapes du flot de conception défini précédemment.

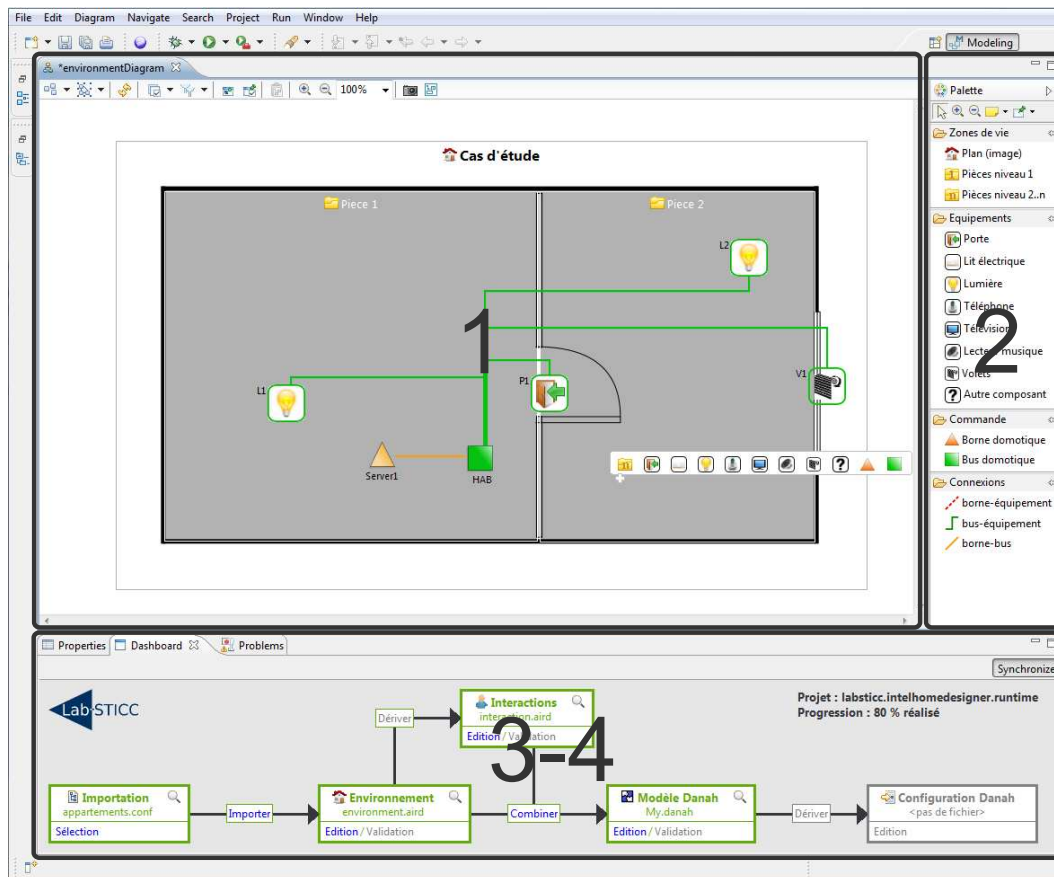


FIGURE 3.6 – Organisation du logiciel *IntelHome Designer* en perspectives

La figure 3.6 donne un aperçu de ce logiciel basé sur l’*IDE (Integrated Development Environment)* Eclipse [Eclipse 2007]. Il est composé des perspectives suivantes :

1. Le **Canevas** permet de définir et de manipuler les éléments graphiques d’un modèle (environnement ou interactions).
2. La **Palette** permet d’importer des éléments graphiques pour les mettre sur le canevas (i.e. les intégrer dans le modèle).
3. La perspective **Properties** permet d’éditer chaque élément sélectionné dans le canevas pour définir/modifier leurs paramètres.
4. La perspective **DashBoard** permet de guider l’utilisateur dans les étapes de la conception : elle intègre un tableau de bord qui reprend les étapes du flot.

Le DSML, formellement défini dans le chapitre précédent, doit aussi être outillé. Pour cela, nous utilisons le langage de méta-modélisation Ecore/EMOF [OMG 2006]. L’édition et la manipulation des modèles se fait à l’aide de *EMF (Eclipse Modeling Frame-*

work) [Steinberg *et al.* 2008] et sa représentation arborescente. Les modèles d'environnement et d'interactions sont quant à eux synchronisés avec une représentation graphique définie à l'aide d'*Obeo Designer* [Obeo 2012]. Les transformations de modèles sont implémentées en utilisant le langage *ATL* [Jouault *et al.* 2006] (*Atlas Transformation Language*). Enfin, la génération de texte (i.e. génération du fichier de configuration Danah à partir de son modèle) est réalisée avec *Acceleo* [Musset *et al.* 2006].

### 3.3.1.2 Tableau de bord

Le tableau de bord de la perspective *Dashboard* a pour but de donner des indications au concepteur sur l'état d'avancement de la conception. Il permet surtout de faciliter l'édition de modèles, leur vérification syntaxique et le passage d'une étape à une autre du flot. L'objectif est donc d'avoir une interface unique pour réaliser ces fonctions (e.g. ne pas avoir besoin de parcourir les menus du plug-in ATL pour lancer une transformation). La figure 3.7 donne un aperçu du tableau de bord développé qui reprend les différentes étapes de la conception domotique :

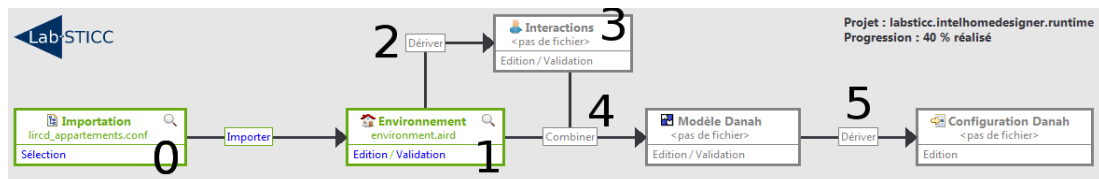


FIGURE 3.7 – Tableau de bord (*Dashboard*) reprenant les étapes de la conception

0. **Importation** du fichier de configuration : le lien *Sélection* permet au concepteur de renseigner un fichier unique (extension “.conf” regroupant les fichiers de configuration à intégrer dans le projet (i.e. fichiers *ETS* et/ou fichiers *LIRC*). Le lien *Importer* permet de parser ces fichiers et d’intégrer les commandes présentes dans le modèle d’environnement.
1. **Modélisation de l’environnement** domotique : le lien *Edition* permet d’afficher le canevas (représentation graphique) du modèle d’environnement pour que le concepteur commence ou reprenne son activité de modélisation. Une fois cette activité terminée, le lien *Validation* lance une vérification syntaxique du modèle défini par rapport à son méta-modèle.
2. **Transformation environnement** → **interactions** : le lien *Dériver* permet de lancer la transformation de modèles “*Env2Env*” pour raffiner le modèle d’environnement, puis la transformation “*Env2Int*” pour générer automatiquement un modèle d’interactions reprenant les opérations du modèle d’environnement.
3. **Modélisation des interactions** utilisateur-système : le lien *Edition* permet d’afficher le canevas pour l’édition du modèle d’interactions. Tout comme dans l’étape 1, le lien *Validation* permet de valider le modèle défini. Dans le cas où ce

dernier n'est pas conforme à son méta-modèle, une fenêtre s'ouvre pour lister les erreurs de modélisation et les modifications à apporter.







4. **Transformation intelhome → danah** : le lien *Combiner* permet de lancer la transformation de modèles "Intel2Danah" pour générer un modèle spécifique à la plateforme Danah à partir des deux modèles définis dans les étapes précédentes. Une fenêtre s'ouvre pour mettre en relation les commandes et labels importés dans l'étape 0 aux opérations et ressources du modèle d'environnement.
5. **Génération du fichier de configuration Danah** : le lien *Dériver* permet finalement de lancer le générateur de code *Acceleo* permettant d'aboutir à un fichier de configuration pour chaque borne domotique supportant le middleware Danah.

Pour indiquer l'état d'avancement, un code couleur est utilisé : les blocs représentant les étapes de la conception qui ont été réalisées sont en couleur verte, les autres restent grisées. Un indicateur sur la progression est également affiché. D'après la figure, elle est estimée à 40% (2/5 blocs fonctionnels). L'édition des modèles associés à chaque bloc fonctionnel déjà traité peut également se faire en lecture seule (e.g. lors de la définition d'un service, le concepteur peut avoir besoin de visualiser l'environnement domotique sans pour autant modifier ce dernier.).

### 3.3.2 Éditeur graphique

#### 3.3.2.1 Modèle d'environnement

Le tableau 3.1 reprend la syntaxe graphique pour le méta-modèle d'environnement :

| Nom (MME <code>Environment</code> )                 | Représentation  |   | Description   |
|---|---|---|---|
|   | Palette   | Canevas   |   |
| Plan ( <code>System</code> )                        |  |  | Plan de l'habitat à modéliser (format png/svg)                                |
| Niveau 1<br>( <code>Aggregate_Component</code> )    |  |  | Composant agrégé de niveau 1 à intégrer directement dans le plan de l'habitat |
| Niveau 2..n<br>( <code>Aggregate_Component</code> ) |  |  | Composant agrégé de niveau 2 ou plus  |











| Nom ( <i>MMEnvironment</i> )               | Représentation  |  | Description   |
|--|---|--|---|
|  | Palette   | Canevas  |   |
| Équipement<br>( <i>Basic_Component</i> )   |  |   | Équipement domotique (Porte, Lit électrique, Lumière, Téléphone, Télévision, etc.) à configurer (couleur associée en fonction du protocole de communication utilisé : noir si aucun protocole, vert pour le <i>KNX</i> , rouge pour l'infrarouge, etc.) |
| Borne domotique ( <i>Server</i> )          |  |   | Serveur domotique responsable de la commande et de la supervision des équipements   |
| Bus domotique ( <i>HA_Bus</i> )            |  |   | Bus domotique responsable de la commande et de la supervision des équipements reliés à une borne domotique  |
| Borne-équipement<br>( <i>server_link</i> ) |  |   | Connexion entre une borne domotique et un équipement (dans notre contexte, il s'agit principalement d'une connexion infrarouge)   |
| Bus-équipement ( <i>hab_link</i> )         |  |  | Connexion entre un bus domotique et un équipement   |

TABLE 3.1 – Syntaxe graphique pour le modèle d'environnement *MMEnvironment*

Cette syntaxe permet de représenter les concepts définis dans le DSML de telle sorte qu'une personne non-experte puisse les manipuler. Ainsi, le système est représenté par un plan que le concepteur peut intégrer en fond d'écran. Les composants agrégés (pouvant représenter des pièces) sont représentés par des zones grisées et transparentes à intégrer au plan de l'habitat ou dans un autre composant agrégé. Les équipements, définis par un icône illustratif, peuvent être reliés à des bornes domotiques pour être directement contrôlés en infrarouge. Ils peuvent aussi être reliés à un bus domotique, lui-même relié à une borne pour le contrôle filaire par bus (Les connexions sont de différentes formes et couleurs en fonction des éléments concernés.).

### 3.3.2.2 Modèle d'interactions

Le tableau 3.2 reprend la syntaxe graphique pour le méta-modèle d'interactions :

| Nom (MMInteraction)                              | Représentation  |   | Description   |
|--|---|---|---|
|  | Palette   | Canevas   |   |
| Profil ( <i>Profil</i> )                         |    |    | Profil utilisateur qui contient toutes les interactions (services et modes)                     |
| Service ( <i>Service</i> )                       |    |    | Interaction de type service permettant d'exprimer la formule "Je veux"                          |
| Mode ( <i>Mode</i> )                             |    |    | Interaction de type mode permettant d'exprimer la formule "J'interdis"                          |
| Activation manuelle ( <i>C1_Manual</i> )         |    |    | Activation sous la demande de l'utilisateur   |
| Demande confirmation ( <i>C2_SemiAutomatic</i> ) |    |    | Activation automatique avec demande de confirmation   |
| Activation automatique ( <i>C3_Automatic</i> )   |    |    | Activation automatique sous condition   |
| Un seul état ( <i>State</i> )                    |   |   | Déclencheur : condition d'activation avec l'état d'une ressource                                |
| Composition normale ( <i>AND</i> )               |  |  | Déclencheur : composition de plusieurs conditions d'activation                                  |
| Composition alternative ( <i>OR</i> )            |  |  | Déclencheur : composition alternative de plusieurs conditions d'activation                      |
| Composition exclusive ( <i>EX</i> )              |  |  | Déclencheur : composition exclusive de plusieurs conditions (i.e. une seule condition vérifiée) |
| Un temps ( <i>Clock</i> )                        |  |  | Déclencheur : condition d'activation par rapport au temps (i.e. heure, plage horraire)          |
| Service élémentaire ( <i>N1_Basic_service</i> )  |  |  | Service élémentaire associé à une opération   |
| Scénario ( <i>N2_Composite</i> )                 |  |  | Composition de services avec prise en compte éventuelle du résultat d'exécution                 |
| Service d'objectif ( <i>N3_HighLevel</i> )       |  |  | Service de haut niveau sémantique associé à une fonction du système                             |



| Nom ( <i>MMInteraction</i> )                               | Représentation |         | Description  |
|--|----------------|---------|--|
|  | Palette        | Canevas |  |
| Interdiction 1<br>( <i>R1_Restriction</i> )                |                |         | Restriction sur un ou plusieurs états d'une ou plusieurs ressources (composants élémentaires)            |
| Interdiction 2<br>( <i>R2_Restriction</i> )                |                |         | Restriction sur un ou plusieurs états d'une ou plusieurs ressources (composants élémentaires et agrégés) |
| Interdiction d'objectif<br>( <i>R3_Restriction</i> )       |                |         | Restriction indépendante de toute ressource associée au concept de super-état                            |
| Activation-action ( <i>activation</i> )                    |                |         | Association entre une condition d'activation et une action (interaction de type service ou mode)         |
| Service alternatif<br>( <i>reconfiguration_service</i> )   |                |         | Définition d'un service alternatif pour prise en compte de l'échec d'un service                          |
| Étape ( <i>Place</i> )                                     |                |         | Étape de composition associée à l'activation d'un service  |
| Transition ( <i>Transition</i> )                           |                |         | Transition permettant de définir une temporisation avant l'activation de chaque service                  |
| Transition-étape<br>( <i>node_source</i> )                 |                |         | Lien entre une transition et une étape : transition → étape  |
| Étape-transition (réussite)<br>( <i>node_target_true</i> ) |                |         | Lien entre une étape et une transition en considérant l'exécution du service correcte                    |
| Étape-transition (échec)<br>( <i>node_target_false</i> )   |                |         | Lien entre une étape et une transition en considérant l'exécution du service incorrecte                  |
| Étape-transition (indiff.)<br>( <i>next_place</i> )        |                |         | Lien entre une étape et une transition sans prendre en compte le résultat d'exécution du service         |
| Marqueur début ( <i>Start</i> )                            |                |         | Marqueur de début obligatoire pour tout scénario   |
| Marqueur de fin ( <i>End</i> )                             |                |         | Marqueur de fin obligatoire pour tout scénario   |

TABLE 3.2 – Syntaxe graphique pour le modèle d'interactions *MMInteractions*

Le profil utilisateur contient toutes les interactions (services et modes) représentées par des boîtes de couleur grise. On retrouve dans chaque interaction un type de condition d'activation (manuelle, automatique ou avec demande de confirmation) et un type d'action (trois niveaux de restriction pour les modes et trois niveaux sémantiques pour les services) représenté par des blocs de couleur verte pour les services, ou rouge pour les modes. Tout comme pour le modèle d'environnement, deux moyens de sélection sont mis à disposition du concepteur : i) le menu contextuel qui propose la création

d'éléments fils de l'élément qui est pointé par la souris et ii) la palette qui regroupe tous les éléments que l'on peut définir dans son modèle. Au dessus du canevas, des filtres et des calques peuvent être ajoutés/retirés pour cacher certains éléments graphiques du canevas ou de la palette. Par exemple, dans l'éditeur graphique de l'environnement domotique, les connexions entre les équipements, les serveurs et les bus domotiques peuvent être cachées pour faciliter la visualisation du modèle.

Le logiciel auquel nous aboutissons permet d'outiller le flot de conception théorique proposé. Il permet aussi de donner au concepteur non-expert les outils pour modéliser l'environnement domotique et les interactions utilisateur-système. Un passage à l'échelle dans les appartements tremplins du centre de Kerpape est présenté dans le chapitre 5 pour illustrer chacune des étapes du flot sur un cas d'étude réel.

### 3.4 Conclusion du chapitre / contribution

Dans ce chapitre, un flot de conception a été présenté. Basé sur l'*IDM*, il permet d'automatiser les différentes étapes de la conception domotique et d'y intégrer le concepteur non-expert ainsi que l'utilisateur dépendant. En effet, à partir de modèles indépendants de toute plateforme (*PIM Environnement* et *PIM Interactions*) définis à l'aide du DSML, la partie commande spécifique à une plateforme domotique peut être générée pour convenir aux besoins de l'utilisateur. Une implémentation possible de ce flot est mise en oeuvre en intégrant le modèle spécifique à la plateforme domotique Danah (*PSM*). Le chapitre suivant montre une deuxième implémentation de ce flot pour la supervision non-intrusive de systèmes domotiques basée sur les ontologies.



# 4

## Modèle de supervision non-intrusif

Semantics is about *“real objects and imaginary concepts and the particular relations between them...”*

Tim Berners-Lee, 1994

La supervision domotique est communément basée sur l'utilisation de capteurs distribués dans l'habitat. Nous pensons que l'analyse sémantique des commandes domotiques envoyées par l'utilisateur fournit des informations suffisamment pertinentes pour assurer la supervision de tels systèmes. Une ontologie est ainsi défini pour modéliser les interactions entre l'utilisateur et son système domotique. À partir de ce modèle de représentation de connaissances largement utilisé dans le web sémantique, nous proposons d'analyser chaque log de commande. L'objectif de ce chapitre est de présenter notre modèle de supervision non-intrusif. Après avoir discuté de la supervision domotique non-intrusive telle que nous la voyons dans ces travaux, nous passons en revue les modèles de supervision existants dans l'état de l'art (section 4.1). Tous ces travaux utilisent des données de capteurs, à partir desquelles des informations contextuelles sont inférées par raisonnement logique. À l'inverse, nos travaux s'intéressent à l'analyse des commandes envoyées par l'utilisateur. L'ontologie et les règles définies permettent tout d'abord de modéliser les interactions entre l'utilisateur et son système domotique en considérant uniquement les logs de commande. L'utilisation d'un moteur d'inférences permet ensuite d'inférer la localisation de l'utilisateur, son intention et d'aboutir ainsi à la proposition de services (section 4.2). Dans le contexte d'un déploiement à échelle réelle, ce modèle de supervision s'intègre enfin dans un flot de conception dirigé par les modèles pour peupler l'ontologie. Une architecture logicielle est de plus définie pour supporter ce modèle à l'exécution (section 4.3).

## Sommaire

---

|         |   |     |
|---------|---|-----|
| 4.1     | Introduction . . . . .                                      | 101 |
| 4.1.1   | Supervision non-intrusive . . . . .                         | 101 |
| 4.1.1.1 | Définition . . . . .  | 101 |
| 4.1.1.2 | Contexte & Enjeux . . . . .                                 | 102 |
| 4.1.2   | Introduction aux ontologies . . . . .                       | 102 |
| 4.1.2.1 | Les Logiques de Description . . . . .                       | 102 |
| 4.1.2.2 | Les ontologies et le langage OWL . . . . .                  | 105 |
| 4.1.3   | État de l'art & Positionnement . . . . .                    | 107 |
| 4.1.3.1 | Ontologies pour les systèmes pervasifs . . . . .            | 107 |
| 4.1.3.2 | État de l'art en domotique . . . . .                        | 107 |
| 4.1.3.3 | Choix retenu . . . . .                                      | 109 |
| 4.2     | Modélisation des interactions utilisateur-système . . . . . | 110 |
| 4.2.1   | Conceptualisation . . . . .                                 | 110 |
| 4.2.1.1 | Contexte d'application . . . . .                            | 110 |
| 4.2.1.2 | Ontologie (T-Box) . . . . .                                 | 110 |
| 4.2.2   | Analyse sémantique . . . . .                                | 111 |
| 4.2.2.1 | Localisation de l'utilisateur . . . . .                     | 111 |
| 4.2.2.2 | Intention de l'utilisateur . . . . .                        | 112 |
| 4.2.2.3 | Proposition de services . . . . .                           | 113 |
| 4.3     | Intégration dans le flot de conception . . . . .            | 114 |
| 4.3.1   | La spécification ODM . . . . .                              | 115 |
| 4.3.1.1 | Les ontologies et l'approche MDA . . . . .                  | 115 |
| 4.3.1.2 | Ontology Definition Metamodel . . . . .                     | 116 |
| 4.3.2   | Flot de conception pour la supervision . . . . .            | 118 |
| 4.3.3   | Architecture de supervision . . . . .                       | 119 |
| 4.4     | Conclusion du chapitre / contribution . . . . .             | 120 |

---

## 4.1 Introduction

Dans un premier temps, nous donnons une définition de la supervision non-intrusive (section 4.1.1) avant d'introduire les ontologies dans la section 4.1.2. Un état de l'art est finalement dressé sur la supervision basée sur les ontologies pour positionner notre contribution par rapport à l'existant (section 4.1.3).

### 4.1.1 Supervision non-intrusive

#### 4.1.1.1 Définition

**Supervision** En automatique, la supervision industrielle est une technique de suivi et de pilotage informatique de procédés de fabrication automatisés [Zamaï *et al.* 2007]. La supervision représente tant l'acquisition des données (e.g. mesures, alarmes, retours d'états de fonctionnement) que l'élaboration des paramètres de commande des processus confiés généralement à des automates programmables. Dans le contexte des systèmes pervasifs, la sensibilité au contexte (*Context-Awareness*) fait référence aux capacités des systèmes électroniques à capter des informations sur leur environnement et à réagir en conséquence. Pour cela, plusieurs fonctions sont nécessaires : i) collecte des informations, ii) détection des changements de situations de contexte et iii) réalisation des adaptations (sans intervention de l'utilisateur). Dans le cadre de ces travaux, le terme supervision fait référence à ces deux dimensions : nous appelons supervision à la fois le monitoring de l'utilisateur (e.g. l'acquisition de données pour la reconnaissance d'activités, la gestion d'alertes) mais aussi la proposition de services. La modélisation et l'interprétation du contexte (i.e. *qui fait quoi et où*) permet ainsi l'adaptation dynamique et contextuelle de l'environnement domotique aux besoins de l'habitant : à partir des données relevées par des capteurs, des services adaptés lui sont proposés.

**Intrusivité** "qui s'introduit de manière non souhaitée" [Demongeot 2005]. L'intrusivité est subjective : perçue par la personne, elle dépend entre autres de son degré de tolérance. Dans le contexte de la domotique, il existe un grand nombre de situations où l'intrusivité peut être perçue différemment suivant l'utilisateur :

- L'utilisation de caméras peut être perçue comme intrusive du fait de la nature des informations enregistrées (flux vidéo permettant de reconstituer précisément les activités de la personne).
- L'intégration physique dans l'environnement domotique de capteurs peut aussi être perçue comme intrusive dans certains cas à cause de la visibilité des équipements (e.g. détecteurs de mouvements infrarouges).
- L'ajout même de contacteurs au niveau des portes, des tiroirs ou des placards peut également être perçu comme intrusif si l'utilisateur en est conscient.

L'intrusivité liée à l'intégration de capteurs dans un habitat n'est pas mesurable objectivement : il est donc difficile de proposer une solution universelle qui réponde à tous

les besoins. Dans ces travaux, le système de supervision ne nécessite pas l'intégration de nouveaux équipements dans le système domotique existant. Pour cette raison, nous pensons que notre modèle de supervision peut être considéré comme peu intrusif.

#### 4.1.1.2 Contexte & Enjeux

Des travaux précédents ont pu mettre en avant l'intérêt d'analyser les commandes domotiques envoyées par l'utilisateur à l'exécution. Dans [Truong *et al.* 2009a], le modèle de supervision proposé considère un service domotique comme un motif qui peut se répéter dans le temps. Basé sur l'hypothèse d'activités très régulières (i.e. logique temporelle), l'objectif est d'extraire automatiquement des scénarios de vie pour automatiser le fonctionnement de l'habitat pour des utilisateurs à mobilité réduite (i.e. en fauteuil roulant électrique). Cependant, les scénarios qui suivent une logique temporelle ne représentent pas tous les services domotiques qui peuvent être proposés à l'utilisateur [De Lamotte *et al.* 2008, Truong 2010]. Par exemple, l'utilisateur a besoin d'allumer ou d'éteindre les lumières dans les pièces qu'il traverse pour aller aux toilettes la nuit. Comme nous l'avons souligné dans les motivations (section 1.3.1), ce genre de services peut être sollicité de manière ponctuelle et non-régulière : à n'importe quelle heure de la nuit, à n'importe quelle journée de la semaine, etc. Contrairement à une analyse fréquentielle/statistique, nous proposons d'utiliser une analyse sémantique de chaque commande de l'utilisateur. Dans cette perspective, il est possible de proposer à l'utilisateur des services adaptés sans aucune phase d'apprentissage. Pour analyser la sémantique associée aux commandes domotiques, nous proposons d'utiliser une ontologie. Basé sur les logiques de description, ce modèle de représentation de connaissances nous permet de définir les interactions entre l'utilisateur et son système domotique de manière formelle et structurée. Un moteur d'inférences interprète ensuite les liens sémantiques de l'ontologie et les règles associées pour déduire la localisation de l'utilisateur et son intention (basé sur un raisonnement logique) et aboutir ainsi à la proposition de services.

### 4.1.2 Introduction aux ontologies

#### 4.1.2.1 Les Logiques de Description

Les *logiques de description* (LD) permettent de représenter la connaissance d'un domaine particulier de manière *formelle* et *structurée* [Baader 2003, Fournier-Viger 2005]. Elles se déclinent en une famille de langages plus ou moins expressifs à partir desquels il est possible de créer une base de connaissances (*Knowledge Base*). L'intérêt principal des LD réside dans la capacité à inférer de nouvelles connaissances à partir de la base de connaissances existante. Cette dernière se divise généralement en deux parties :

- **T-Box** : taxonomie, structure hiérarchique de concepts (*Intensional Knowledge*)
- **A-Box** : instances, individus particuliers (*Extensional Knowledge*)

| T-Box   | A-Box                                    |
|---|--|
| $Femelle \sqsubseteq \top \sqcap \neg Male$             | $Humain(Claire)$                         |
| $Male \sqsubseteq \top \sqcap \neg Femelle$             | $Femelle(Claire)$                        |
| $Animal \equiv Male \sqcup Femelle$                     | $Femme(Claudine)$                        |
| $Humain \sqsubseteq Animal$                             | $Humain(Bernard)$                        |
| $Femme \equiv Humain \sqcap Femelle$                    | $\neg Femelle(Bernard)$                  |
| $Mere \equiv Femme \sqcap \exists relationParentEnfant$ | $relationParentEnfant(Claudine, Claire)$ |

TABLE 4.1 – Extraits d’une base de connaissances : T-Box à gauche / A-Box à droite

Les informations représentant la base de connaissances sont modélisées par un couple  $(T, A)$ , où  $T$  est un ensemble de formules relatives aux informations terminologiques (la T-Box) et  $A$  est un ensemble de formules relatives aux informations sur les assertions (la A-Box). Les LD utilisent alors les notions de *concepts*, de *rôles* et d'*individus* pour parler des éléments d’une base de connaissances (cf tableau 4.1) :

- les **concepts** sont des classes d’éléments, interprétés comme un ensemble dans un univers donné (e.g. *Humain*). Les LD prédéfinissent minimalement deux concepts atomiques :  $\top$  représente les concepts les plus généraux de la base de connaissances et  $\perp$  représente les concepts les plus spécifiques (ensemble vide).
- les **rôles** correspondent aux liens entre les éléments, interprétés comme des relations binaires sur un univers donné (e.g. *relationParentEnfant*).
- les **individus** correspondent aux éléments d’un univers donné (e.g. *Claire*)

Pour décrire, lier et interpréter ces éléments, différentes notions sont définies :

- Une **interprétation** permet tout d’abord d’explicitement formellement la sémantique d’une T-Box. Une interprétation  $\chi$  est composée d’un domaine d’interprétation  $\Delta^\chi$  représentant un ensemble d’individus et d’une fonction d’interprétation  $^\chi$  qui assigne à chaque concept atomique  $A$  un ensemble tel que  $A^\chi \subseteq \Delta^\chi$ .
- Les **axiomes terminologiques** permettent de définir des relations entre les concepts. On y retrouve entre autres des relations d’équivalence (e.g.  $C \equiv D$ ) ou d’inclusion (e.g.  $C \sqsubseteq D$ ). Une interprétation doit satisfaire ces axiomes (e.g.  $C^\chi = D^\chi$  et  $C^\chi \subseteq D^\chi$ ).
- Les **assertions** s’expriment en terme de concepts et rôles (T-Box) sur les individus (e.g. *Claire* et *Claudine*). Une interprétation  $\chi$  satisfait une A-Box  $A$  si et seulement si  $\chi$  satisfait toutes les assertions de  $A$ . Il existe deux types d’assertion : i) les *assertions d’appartenance* :  $Humain(Claire)$  et ii) les *assertions de rôle* (e.g.  $RelationParentEnfant(Claudine, Claire)$ )
- Des **constructeurs** permettent aux concepts et aux rôles d’être combinés pour former des concepts et rôles composés (ex :  $Humain \sqcap Femelle$ ). Le nombre de ces constructeurs varie selon le langage de LD considéré.



### LA LOGIQUE MINIMALE $\mathcal{AL}$

Le langage de description  $\mathcal{AL}$  représente la logique minimale à partir de laquelle dérive tous les autres langages de description [Schmidt-Schaufl & Smolka 1991]. Le tableau ci-dessous présente les constructeurs de ce langage :

| Constructeur | Description                   |
|--------------|-------------------------------|
| $A$          | un concept atomique           |
| $\top$       | le concept universel          |
| $\perp$      | le concept le plus spécifique |
| $\neg$       | la négation atomique          |
| $C \sqcap D$ | l'intersection                |

Il existe plusieurs extensions de  $\mathcal{AL}$  qui introduisent de nouveaux constructeurs (union, cardinalité, etc.). Malgré tout, la majorité des langages dérivés peut être traduite en utilisant la logique  $\mathcal{AL}$ . L'équation 4.1 montre la traduction du constructeur d'union  $\sqcup$  avec les constructeurs de  $\mathcal{AL}$ , à savoir la négation atomique  $\neg$  et l'intersection  $\sqcap$ .

$$C \sqcup D \equiv \neg(\neg C \sqcap \neg D) \quad (4.1)$$

**Les inférences** L'inférence s'effectue au niveau terminologique comme au niveau factuel (instances). Les relations exploitées sont essentiellement des relations de **subsumption** pour la définition d'une structure hiérarchique et des relations d'**instance** entre concept et individu.

L'inférence au niveau terminologique comprend la résolution des types de propriétés suivants :

- **Satisfiabilité** : un concept  $C$  est satisfiable si et seulement si il existe un modèle (cf interprétation)  $\chi$  tel que  $C^\chi \neq \emptyset$
- **Subsumption** : un concept  $C$  est subsumé par un concept  $D$  si et seulement si  $C^\chi \subseteq D^\chi$  pour tout modèle  $\chi$
- **Équivalence** : un concept  $C$  est équivalent à un concept  $D$  si et seulement si  $C^\chi = D^\chi$  pour chaque modèle  $\chi$
- **Disjonction** : des concepts  $C$  et  $D$  sont disjoints si et seulement si  $C^\chi \cap D^\chi = \emptyset$  pour tout modèle  $\chi$

L'inférence au niveau factuel comprend la résolution des types de propriétés suiv-

ants :

- **Cohérence** : une A-Box  $A$  est cohérente par rapport à une T-Box  $\mathcal{T}$  si et seulement s'il existe un modèle  $\chi$  de  $A$  et  $\mathcal{T}$ .
- **Vérification d'instance** : Vérifier par inférence si une assertion  $C(a)$  est vraie pour tout modèle  $\chi$  d'une A-Box  $A$  et d'une T-Box  $\mathcal{T}$ .
- **Vérification de rôle** : Vérifier par inférence si une assertion  $R(a, b)$  est vraie pour tout modèle  $\chi$  d'une A-Box  $A$  et d'une T-Box  $\mathcal{T}$ .
- **Problèmes de récupération** : Pour une A-Box  $A$ , un concept  $C$  d'une terminologie  $\mathcal{T}$ , inférer les individus  $a^x1...a^xn \in C^x$  pour tout modèle  $\chi$  de  $\mathcal{T}$ .

Les logiques de description reposent donc sur un langage formel qui se distingue des autres langages de description ou de modélisation (e.g. UML) de part les caractéristiques suivantes : i) une **sémantique formelle** basée sur la logique et ii) la capacité d'**inférer** des connaissances implicites à partir d'une base de connaissances (explicite).

**Hypothèse du monde ouvert** L'interprétation des connaissances respecte l'hypothèse du *monde ouvert* où l'absence d'informations représente l'ignorance plutôt qu'une information négative. Par exemple, l'assertion  $relationParentEnfant(Claudine, Claire)$  se prête à plusieurs interprétations. Alors que les BDD interprètent cette information comme "*Claudine n'a qu'une seule fille*", les LD interprètent cette information comme "*Claire est une fille de Claudine*". Ainsi, les LD ne ferment pas la possibilité que Claire ait une soeur dans un modèle lié à une interprétation particulière (une A-Box peut avoir plusieurs interprétations) : elles laisse la possibilité d'inférer que "*Claudine a plus d'une fille*". Cela oppose le raisonnement logique lié aux logiques de description à la vérification de modèles finis (*finite model checking*) associée aux BDD. Les LD utilisent une approche ontologique : "pour décrire les individus d'un domaine, elles requièrent tout d'abord la définition des catégories générales d'individus et les relations logiques que les individus ou catégories peuvent entretenir entre eux" [Fournier-Viger 2005].

#### 4.1.2.2 Les ontologies et le langage OWL

Une ontologie est une représentation formelle et structurée d'un domaine de connaissances [Baader 2003] Elle permet en ce sens de modéliser un ensemble de concepts, ainsi que leurs relations sous une forme exploitable par une machine. L'intérêt d'une telle représentation réside dans l'automatisation des processus de raisonnement et de décision, appelés processus cognitifs. Selon Gruber, une ontologie est "*la spécification d'une conceptualisation d'un domaine de connaissances*". Cette définition porte deux dimensions [Gruber *et al.* 1995] :

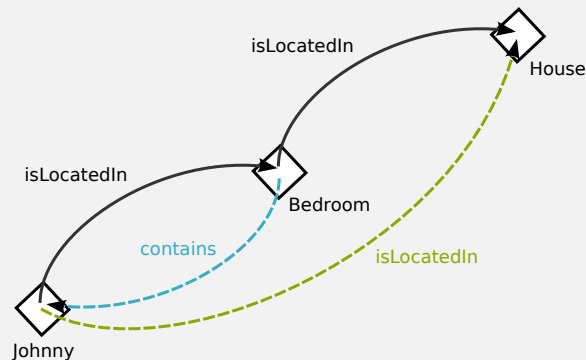
1. une ontologie est la **conceptualisation** d'un domaine, c'est à dire un *choix* quant à la manière de décrire un domaine
2. c'est par ailleurs la **spécification** de cette conceptualisation, c'est à dire sa *description formelle*, basée sur les logiques de description (cf section 4.1.2.1)

En plus de la modélisation conceptuelle, le deuxième point souligne l'importance du langage de spécification dans la construction d'une ontologie. En effet, basé sur la sémantique formelle des logiques de description, il permet de représenter les concepts ainsi que la nature des liens entre ces concepts. *OWL (Ontology Web Language)* est le langage le plus utilisé pour représenter les ontologies [McGuinness *et al.* 2004]. Des règles *SWRL (Semantic Web Rule Language)* [Horrocks *et al.* 2004] peuvent être ajoutées à l'ontologie pour compléter sa description. Il s'agit d'une recommandation du *W3C (World Wide Web Consortium)* qui combine le langage *OWL* avec celui du *Rule Markup Language*. Un moteur d'inférences peut alors interpréter les liens sémantiques de l'ontologie et les règles associées pour inférer de nouvelles connaissances.

*OWL* est basé sur la syntaxe *RDF (Resource Description Framework)* pour l'organisation des données et méta-données en un ensemble de triplets de la forme {**sujet**, **prédicat**, **objet**} où le **sujet** représente l'entité à décrire, le **prédicat** représente le type de propriétés associé à cette ressource et l'**objet** représente la valeur de cette propriété (e.g. {Johnny, *isLocatedIn*, Bedroom}). Par rapport à *RDF*, *OWL* (qui adopte une approche orientée objet) ajoute les concepts de classe et de propriété d'équivalence, d'égalité, de différence, d'inverse, de symétrie, de transitivité, etc. Une ontologie prend finalement la forme d'un graphe composé de classes (i.e. *T-Box*) et d'individus (i.e. *A-Box*) dont les relations peuvent être principalement de deux types : 1) des relations sémantiques de différentes natures et 2) des relations de subsomption (i.e. inclusion).

### ILLUSTRATION DES PROPRIETES OWL

Les relations sémantiques entre concepts, définies par les *ObjectProperties* en *OWL*, peuvent être de natures différentes. La figure ci-dessous illustre les propriétés inverse (en bleu) et de transitivité (en vert) qui peuvent être déduites par un moteur d'inférences (liens en pointillés) :



Legend: ◇ owl:Individual → owl:ObjectProperty --- inferred owl:ObjectProperty

Prenons l'exemple de la propriété *isLocatedIn*, définie comme transitive. Le moteur d'inférences, sachant que  $\{Johnny, isLocatedIn, Bedroom\}$  et que  $\{Bedroom, isLocatedIn, House\}$ , peut inférer que  $\{Johnny, isLocatedIn, House\}$ .

### 4.1.3 État de l'art & Positionnement

#### 4.1.3.1 Ontologies pour les systèmes pervasifs

On peut distinguer trois enjeux quant à l'utilisation des ontologies dans le domaine de la domotique, et plus généralement dans celui des systèmes pervasifs :

1. L'**interopérabilité** : les ontologies permettent entre autres la représentation de concepts domotiques en s'abstrayant des spécificités technologiques, facilitant ainsi le partage de la connaissance par des agents logiciels.
2. Le **raisonnement** : les ontologies définissent des liens sémantiques entre concepts à partir desquels il est possible d'exploiter plusieurs mécanismes de raisonnement logique (i.e. inférences logiques). À la différence de certains domaines d'application ou tout doit être connu, la connaissance sur le contexte d'un environnement domotique impliquant un utilisateur peut être incomplète ou incertaine (cf *Hypothèse du monde ouvert* de la section 4.1.2.1).
3. La **réutilisation** : les ontologies sont couramment définies à l'aide de langages standardisés par le W3C, offrant ainsi la possibilité de les réutiliser. Par exemple, FOAF (*Friend Of A Friend*) [Brickley & Miller 2000] est une description sémantique qui décrit des personnes, leurs caractéristiques et les relations d'amitié entre ces personnes. DAML-Time [Hobbs *et al.* 2002] est une ontologie qui traite des connaissances générales sur le temps et les événements temporels.

#### 4.1.3.2 État de l'art en domotique

Certains travaux s'intéressent aux ontologies pour garantir l'interopérabilité du système domotique. Dans [Reinisch *et al.* 2009], une ontologie est définie pour modéliser un environnement domotique à grande échelle (*Building Automation Systems*). Elle permet de faire face à l'hétérogénéité des systèmes domotiques issus de technologies différentes (*KNX, LonWorks, BACnet*). Au lieu d'utiliser des passerelles entre ces différentes plateformes (qu'il faut configurer avec des outils spécifiques), l'ontologie représente une abstraction de l'infrastructure réseau hétérogène. DogOnt (*Domotic OSGI Gateway*) [Bonino & Corno 2008, Bonino *et al.* 2009] supporte également la description d'un habitat de manière indépendante du matériel et des télécommunications utilisés. L'application de règles SWRL et une reconnaissance automatique de classes d'équipements par inférences logiques permettent par la suite d'attribuer les paramètres de configuration matériels et réseaux.

Dans ces travaux, nous utilisons l'IDM pour permettre de garantir l'interopérabilité d'un système domotique. En effet, nous pensons qu'elle apporte des solutions pertinentes pour notre flot de conception. En outre, nous avons pu bénéficier de l'existence d'outils permettant la création rapide d'éditeurs graphiques, la mise en place de transformations de modèles et la vérification syntaxique de ces derniers.

Les mécanismes de raisonnement logique associés aux ontologies jouent un rôle important pour la supervision domotique. Les inférences logiques réalisées par un moteur d'inférences permettent de déduire des connaissances implicites à partir de l'ontologie (modèle explicite) et de règles s'y appliquant. La plupart des travaux portant sur la reconnaissance d'activités basée sur les ontologies reposent sur l'utilisation de triplets *{sujet, verbe, objet}* pour décrire les informations de contexte. *CoBra* [Chen *et al.* 2005] et *Gaia* [Román *et al.* 2002, Bittencourt *et al.* 2009] sont deux des logiciels les plus connus, basés sur une architecture multi-agents. Chaque agent est ainsi chargé de collecter et de traiter les informations contextuelles pour aboutir à la proposition de services. Dans [Román *et al.* 2002], lorsque le nombre de personnes présentes dans une pièce est supérieure à trois et que l'application Powerpoint est active, le système déduit que l'activité de cette pièce est une présentation :

$$\begin{aligned} & \text{People}(\text{Room2401}, \geq, 3) \text{ and } \text{Application}(\text{PowerPoint}, \text{Running}) \\ & \Rightarrow \text{RoomActivity}(2401, \text{Presentation}) \end{aligned} \quad (4.2)$$

Dans [Liao & Tu 2007], le triplet RDF communément utilisé est étendu en prenant en considération un quatrième uplet "*temps*" pour mettre en évidence des situations anormales en interprétant les activités des habitants en fonction du temps écoulé. En fonction des profils d'utilisation temporelle des équipements, des alertes peuvent être générées suivant l'utilisation que l'on en fait. Par exemple, *Device(gas\_switch#1, is-on, 18 : 38 : 41)* permet de décrire dans l'ontologie TempCRM l'ouverture du gaz qui doit être refermé au bout d'un certain temps.

L'implémentation d'une telle architecture reste à l'heure actuelle un processus qui n'est pas trivial du fait de la taille de l'ontologie de contexte qui peut parfois être importante. Conon [Wang *et al.* 2004] sépare en deux parties la modélisation du contexte : la première s'intéresse aux concepts décrivant le système à un haut niveau d'abstraction alors que la deuxième reprend les concepts spécifiques à un domaine. Le concept générique "*IndoorSpace*" se décline par exemple en concepts "*Building*", "*Room*", "*Corridor*" dans le domaine de la domotique. Cette décomposition hiérarchique se retrouve dans [Gouin-Vallerand *et al.* 2012] qui propose une séparation *micro/macro* du contexte pour faciliter l'implémentation d'une telle architecture. L'approche permet ainsi de gérer la complexité relative au traitement d'informations de contexte, en distribuant le calcul sur des *Devices Nodes* (qui ont connaissance du contexte *micro*) sans perdre le bénéfice du traitement au niveau *macro* à l'aide des *Coordinator Nodes*.

Tous ces travaux se basent sur des données issues de capteurs, à partir desquelles des informations contextuelles sont inférées par raisonnement logique. Dans ces travaux, nous proposons également d'utiliser une ontologie, non pas pour analyser les données de capteurs, mais pour analyser la sémantique des logs de commande (requêtes de l'utilisateur). Cela nécessite la définition d'une ontologie modélisant les interactions utilisateur-système qui se distinguent de ces propositions.

#### 4.1.3.3 Choix retenu

Nous pensons que l'analyse des commandes domotiques peut fournir suffisamment d'informations pertinentes (e.g. localisation, activités de l'utilisateur) pour la supervision d'un système domotique.

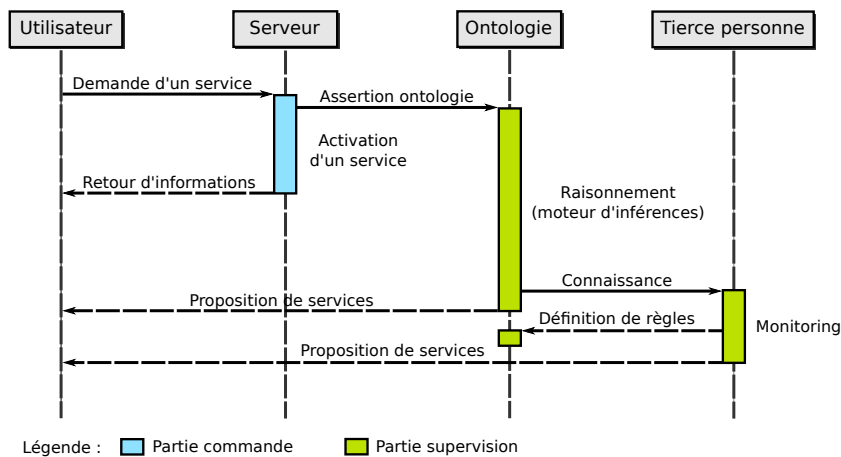


FIGURE 4.1 – Utilisation de l'ontologie pour la supervision non-intrusive

Dans un premier temps, une ontologie permet de modéliser les interactions entre l'utilisateur et son système domotique. Pour permettre l'analyse sémantique des commandes, des règles sont définies pour permettre à un moteur d'inférences de déduire la localisation et l'intention de l'utilisateur et d'aboutir à la proposition de services contextuelle et dynamique. L'utilisation de l'ontologie pour la supervision d'interactions non-intrusive d'un habitat domotisé est illustrée sur le diagramme de séquences de la figure 4.1. Lorsque l'utilisateur sollicite un service domotique, la commande émise est traitée par le serveur domotique, puis une information est retournée à l'utilisateur concernant la réussite ou non de cette activation. La requête de l'utilisateur est également récupérée et traduite en assertion dans l'ontologie de contexte. Un moteur d'inférences est ensuite chargé d'interpréter cette assertion et d'enrichir la base de connaissances :

1. d'une ou plusieurs informations de plus haut niveau que la simple activation de services, que la personne référente (ergothérapeute, membre de la famille, etc.) peut analyser pour définir de nouvelles règles ou services

2. d'une proposition de services sur l'interface de l'utilisateur basée sur les informations précédemment inférées

## 4.2 Modélisation des interactions utilisateur-système

L'utilisation d'ontologies permet d'aboutir à une modélisation fine des interactions utilisateur-système (section 4.2.1). À partir de cette modélisation et de règles associées, il est possible d'inférer la localisation et l'intention de l'utilisateur, aboutissant à la proposition de services (section 4.2.2).

### 4.2.1 Conceptualisation

#### 4.2.1.1 Contexte d'application

La modélisation des interactions utilisateur-système en utilisant le moins de capteurs physiques possible a pour but de réaliser du monitoring d'activités non-intrusif et d'aboutir à la proposition de services. Pour cela, il convient de poser le cadre applicatif du modèle de supervision proposé :

1. L'utilisateur fait une **utilisation logique** de son système domotique : l'ontologie, basée sur les logiques de description, a été définie en considérant ce fait.
2. L'utilisateur est **seul** dans son habitat, équipé d'un terminal mobile (e.g. *smartphone*, *netbook*, *PDA*). Dans notre contexte, il n'est pas possible d'associer un utilisateur à l'activation d'un interrupteur. Nous supposons alors que seuls les logs de commande provenant de cette interface sont analysés<sup>1</sup>.

Le raisonnement logique permet d'aboutir à la proposition dynamique et contextuelle de services. Il ne s'agit pas d'une proposition unique (qui nécessite une résolution optimale) mais d'une liste de services qui peut être mise au premier plan de l'IHM de l'utilisateur en charge de la prise de décision.

#### 4.2.1.2 Ontologie (T-Box)

L'ontologie est constituée d'une partie comprenant des concepts génériques qui peuvent s'appliquer à tous les types d'environnements pervasifs (*Upper Ontology*) qui se déclinent en un ou plusieurs concepts spécifiques au domaine de la domotique (*Specific Ontology*). Elle comprend 6 concepts fondamentaux (cf figure 4.2) :

1. **User** : personne qui interagit avec son environnement via un terminal mobile
2. **Device** : ressource activable i) à distance ou ii) localement (dans la même pièce)

---

1. Ces travaux peuvent s'appliquer à un contexte multi-utilisateurs à condition que chaque commande soit associée à un *id* utilisateur (e.g. services sollicités à partir d'interfaces individuelles).

3. **Basic Service** : fourni par les ressources d'activation locale ou distante
4. **System** : interface pour collecter les interactions (ressources et services)
5. **Location** : contient des ressources et des utilisateurs (e.g. une pièce)
6. **Effect** : effet de l'activation de services sur l'environnement (e.g. luminosité)

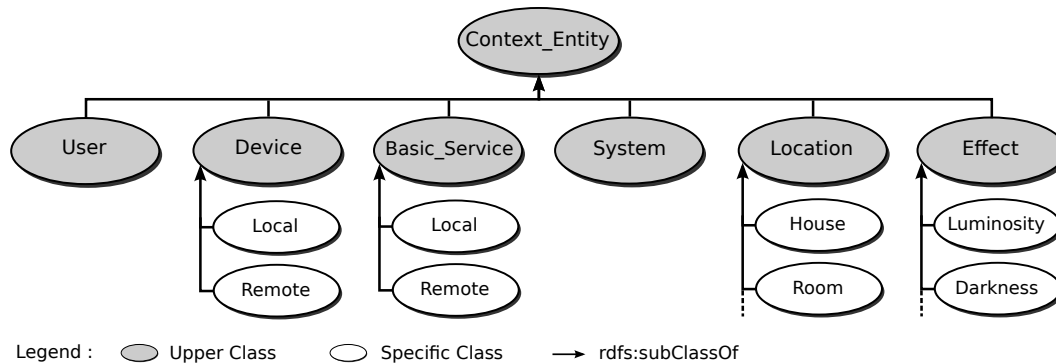


FIGURE 4.2 – Extrait de la taxonomie de l'ontologie (T-Box)

Les concepts présentés ci-dessus sont le support de la supervision non-intrusive. En effet, selon la manière dont l'utilisateur (*User*) interagit avec son système domotique (*System*), nous cherchons à inférer sa localisation (*Location*) et son intention basée sur l'effet (*Effect*) produit par les ressources d'activation locale (*Local\_Device*) lorsqu'il demande un service (*Basic\_Service*). Sachant où se trouve l'utilisateur et quelle est son intention, il est par la suite possible de lui proposer des services adaptés. Cela est rendu possible par la sémantique formelle des logiques de description sous-jacentes aux ontologies mais aussi par les capacités de raisonnement basées sur les règles.

## 4.2.2 Analyse sémantique

### 4.2.2.1 Localisation de l'utilisateur

La localisation est une information de contexte importante, sinon la plus importante, à partir de laquelle des activités peuvent être inférées. Cette information est généralement déduite de données issues de capteurs (e.g. détecteurs de mouvements, contacteurs). Dans nos travaux, nous considérons les activations de services de l'utilisateur. La figure 4.3 montre une partie de la terminologie (*T-Box*) avec les concepts et les liens sémantiques associés pour la localisation de l'utilisateur. Les ressources d'activation locale **LocalDevice** et les ressources d'activation à distance **RemoteDevice** (concepts mutuellement exclusifs) sont définies pour représenter i) les ressources qui sont activées lorsque l'utilisateur se trouve dans la même pièce et ii) les ressources qui sont activées à distance (e.g. à partir d'une pièce voisine). Le concept porte (**Door**) est défini



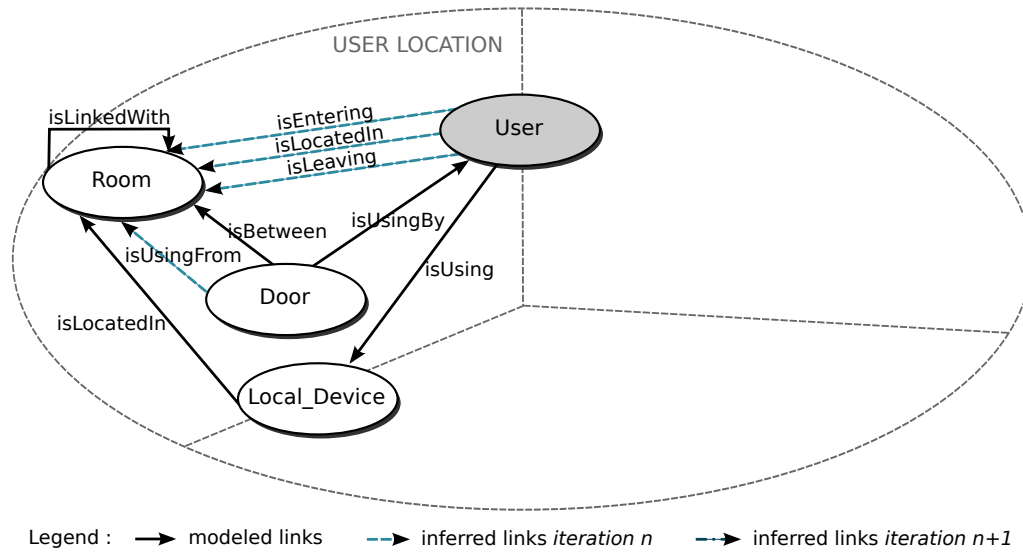


FIGURE 4.3 – Extrait de l’ontologie pour la localisation de l’utilisateur

| Num. | Property    | Expression SWRL ( <i>Semantic Web Rule Language</i> )   |
|------|-------------|---|
| 1    | isLocatedIn | LocalDevice(?d), Room(?r), User(?u), isLocatedIn(?d,?r), isUsing(?u,?d) $\rightarrow$ isLocatedIn(?u,?r)  |
| 2    | isUsingFrom | Door(?d), Room(?r), User(?u), isLocatedIn(?u,?r), isUsingBy(?d,?u) $\rightarrow$ isUsingFrom(?d,?r)   |
| 3    | isLeaving   | Door(?d), Room(?r), User(?u), isUsing(?u,?d), isUsingFrom(?d,?r) $\rightarrow$ isLeaving(?u,?r)   |
| 4    | isEntering  | Door(?d), Room(?r1), Room(?r2), User(?u), isBetween(?d,?r1), isBetween(?d,?r2), isLeaving(?u,?r1), isLinkedWith(?r1,?r2) $\rightarrow$ isEntering(?u,?r2) |

TABLE 4.2 – Règles SWRL pour la localisation de l’utilisateur

avec deux liens **isBetween** vers exactement deux concepts **Room**. À partir de cette représentation logique, des règles sont exprimées pour inférer des propriétés comme **isLocatedIn**, **isEntering**, et **isLeaving** suivant l’activation de services. Considérons par exemple la règle 1 du tableau 4.2 : un utilisateur  $?u$  qui utilise une ressource d’activation locale  $?d$  qui se trouve dans une pièce  $?r$ , se trouve lui même dans cette pièce.

#### 4.2.2.2 Intention de l’utilisateur

Lorsque l’utilisateur demande l’activation d’un service, son objectif est de produire l’effet associé à cette activation sur son environnement. Dans nos travaux, nous cher-

chons à inférer l'effet souhaité par l'utilisateur à partir des commandes sollicitées.

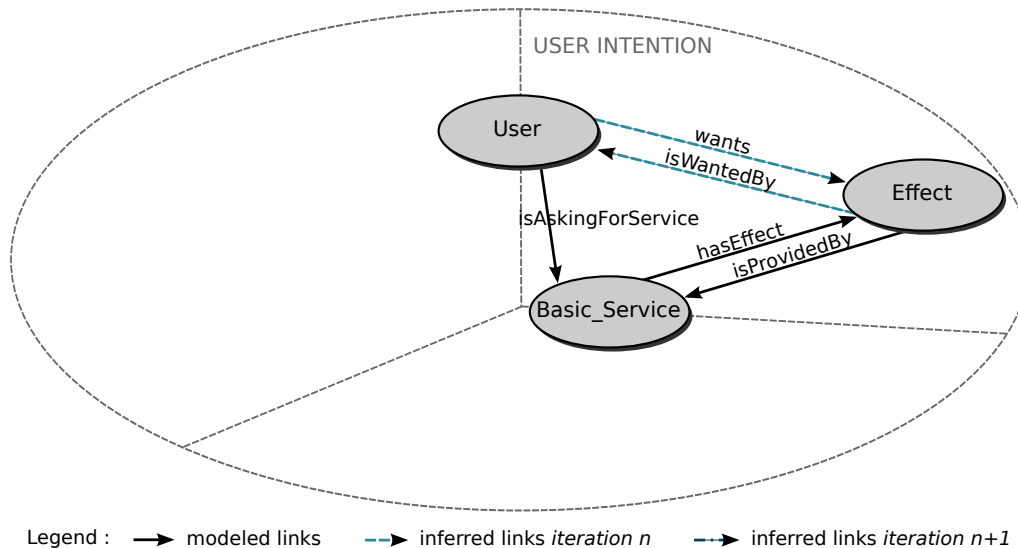


FIGURE 4.4 – Extrait de l'ontologie pour l'intention de l'utilisateur

| Num. | Property | Expression SWRL ( <i>Semantic Web Rule Language</i> )  |
|------|----------|--|
| 5    | wants    | BasicService(?s), Effect(?e), User(?u), hasEffect(?s,?e), isAskingForService(?u,?s) → wants(?u,?e) |

TABLE 4.3 – Règle SWRL pour l'intention de l'utilisateur

La figure 4.4 montre une partie de l'ontologie associée à l'intention de l'utilisateur. Tous les services élémentaires (**BasicService**) ont une propriété d'effet principal (e.g. **Luminosity, Darkness, Heat, Quietness**, etc.) qui leur est associée à travers la relation **hasEffect**. L'intention de l'utilisateur est inférée à partir de cette propriété. Considérons la règle 5 du tableau 4.3 : un utilisateur  $?u$  qui demande l'activation d'un service élémentaire  $?b$ , veut en réalité l'effet  $?e$  associé à ce service.

#### 4.2.2.3 Proposition de services

À partir des informations sur la localisation et l'intention de l'utilisateur inférées précédemment, le prend les éléments présentés précédemment et le joint au concept **System** qui sert d'interface pour la proposition de services. En couplant l'intention de l'utilisateur avec les informations de localisation, il est possible d'aboutir à la proposition de services. Par exemple, un utilisateur qui veut de la lumière (e.g. **wants(User1, Luminosity)**) et qui entre dans une pièce se verra proposer des services permettant d'augmenter la luminosité de la pièce où il entre (à travers l'utilisation des propriétés

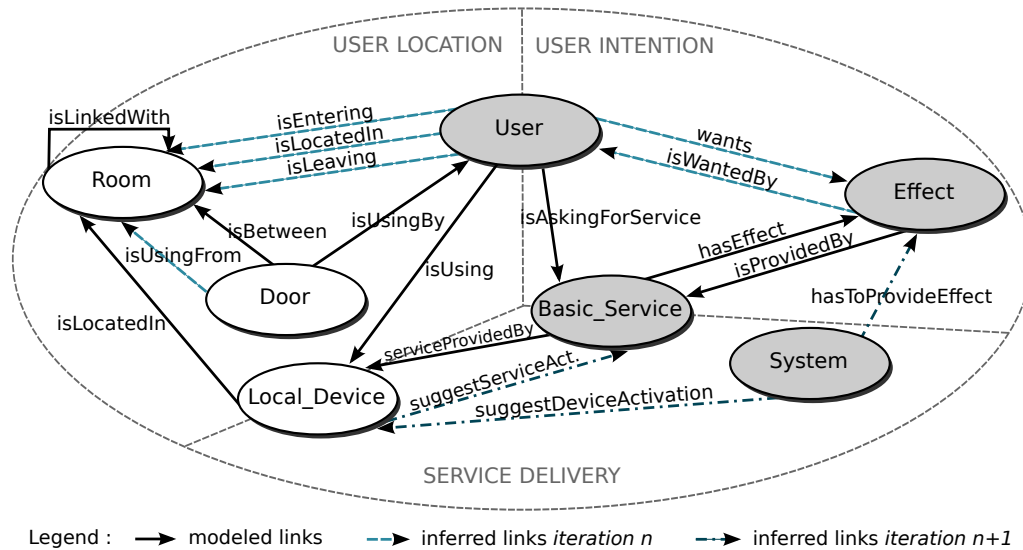


FIGURE 4.5 – Extrait de l'ontologie : localisation, intention et proposition de services

| Num. | Property                 | Expression SWRL ( <i>Semantic Web Rule Language</i> )   |
|------|--------------------------|---|
| 6    | isUsingFrom              | Door(?d), Room(?r), User(?u), isLocatedIn(?u,?r), isUsingBy(?d,?u) $\rightarrow$ isUsingFrom(?d,?r)   |
| 7    | isWantedIn               | Effect(?e), Room(?r), User(?u), isEntering(?u,?r), isWantedBy(?e,?u) $\rightarrow$ isWantedIn(?e,?r)  |
| 8    | suggestDeviceActivation  | BasicService(?b), Device(?d), Effect(?e), Room(?r), System(?s), hasToProvide(?s,?e), isLocatedIn(?d,?r), isProvidedBy(?e,?b), isWantedIn(?e,?r), serviceProvidedBy(?b,?d) $\rightarrow$ suggestDeviceActivation(?s,?d)  |
| 9    | suggestServiceActivation | BasicService(?b), Device(?d), Effect(?e), Room(?r), System(?s), hasToProvide(?s,?e), isLocatedIn(?d,?r), isProvidedBy(?e,?b), isWantedIn(?e,?r), serviceProvidedBy(?b,?d) $\rightarrow$ suggestServiceActivation(?d,?b) |

TABLE 4.4 – Règles SWRL pour la proposition de services

**suggestDeviceActivation** et **SuggestServiceActivation** qui s'appliquent respectivement aux concepts *System* et *Device* (cf tableau 4.3)).

### 4.3 Intégration dans le flot de conception

Le peuplement d'une telle ontologie doit être flexible pour s'adapter aux besoins de l'utilisateur et aux environnements domotiques qui sont amenés à évoluer. En uti-

lisant la spécification ODM (section 4.3.1), nous proposons d'automatiser la génération d'individus (i.e. instances) en intégrant l'ontologie dans le flot de conception dirigé par les modèles (section 4.3.2). Dans la perspective d'un déploiement de services à grande échelle, nous définissons de plus une architecture logicielle pour l'exécution du modèle de supervision (section 4.3.3).

### 4.3.1 La spécification ODM

#### 4.3.1.1 Les ontologies et l'approche MDA

L'IDM et en particulier l'approche MDA est un paradigme émergeant dans le domaine du génie logiciel. Les ontologies sont quant à elles largement utilisées dans le domaine du web sémantique. Si ces deux espaces technologiques ont évolué en parallèle au sein de deux communautés différentes, il existe malgré tout un certain nombre de convergences. *UML* et *OWL* sont les langages (issus respectivement de l'IDM et du Web Sémantique) les plus utilisés et répandus dans ces espaces technologiques. Ils présentent de nombreux points communs : l'aboutissement à une structure hiérarchique (taxonomie) de données grâce à l'héritage (*subsumption*) ou la notion d'instanciation, de peuplement d'*individus* à partir d'un modèle générique.

Il existe cependant des particularités pour chaque espace de modélisation. MDA permet de manipuler un modèle métier dont la syntaxe est précisée dans son métamodèle ("tout modèle est conforme à son métamodèle"). Le langage qui dérive de ce métamodèle comprend alors les constructions syntaxiques ainsi que les relations entre ces constructions. En revanche, l'approche MDA offre peu de moyens pour permettre une interprétation sémantique formelle de ces modèles. Au contraire, les ontologies permettent de manipuler des connaissances par des machines en rendant explicite et en formalisant les relations sémantiques entre concepts. Cela est rendu possible grâce à la sémantique formelle des logiques de description sur lesquelles les ontologies sont basées, ainsi qu'aux possibilités de raisonnement associées (cf section 4.1.2.1).

L'unification de ces deux espaces technologiques semble alors pertinente pour tirer profit de leurs avantages respectifs, et offrir ainsi une modélisation plus complète. Les connaissances d'un modèle métier peuvent être à la base du processus de création d'une ontologie [Hillairet *et al.* 2008]. En effet, avec l'émergence de l'IDM dans le domaine du génie logiciel, la connaissance liée à un domaine particulier est de plus en plus souvent représentée par des modèles métier. Ces abstractions capitalisent un ensemble de connaissances, exprimées par un langage de modélisation (généraliste ou dédié). La réutilisation des connaissances issues de l'expertise métier permet alors d'alléger le processus de construction d'une ontologie. Un processus d'annotation est néanmoins essentiel pour i) enrichir sémantiquement le modèle source (sémantique formelle interprétable par des machines) et ii) sélectionner les concepts importants pour la création d'une base de connaissances (certaines classes du modèle métier n'apportent pas de

valeur ajoutée en terme de connaissances). La base de connaissances peut par la suite être peuplée avec les instances du modèle métier.

Une solution pertinente pour unifier ces deux espaces de modélisation est de rendre conforme OWL (langage d'ontologie le plus utilisé et largement répandu dans le web sémantique) aux trois niveaux de modélisation (M1/M2/M3) de l'approche MDA. Cela permet d'aboutir à une structure commune aux deux langages représentatifs des deux espaces technologiques (i.e. UML et OWL). Cette solution a ainsi été retenue par consensus et aboutit à la spécification ODM.

#### 4.3.1.2 Ontology Definition Metamodel

ODM (*Ontology Definition Metamodel*) est une spécification de l'OMG (*Object Management Group*) [Colomb *et al.* 2006, Consulting & Deere 2005] pour rendre les concepts du MDA applicables à ceux du web sémantique. Elle spécifie des méta-modèles pour différents langages de représentation des connaissances (e.g. RDF et OWL), permettant de passer d'un espace technologique à un autre par transformation de modèles. Ces méta-modèles sont conformes au *MOF* pour en garantir l'interopérabilité.

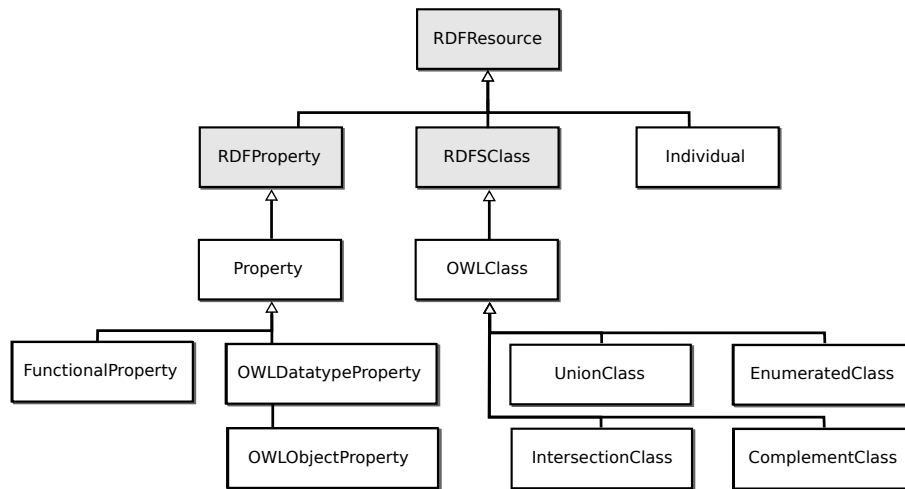


FIGURE 4.6 – Extrait du méta-modèle OWL (taxonomie)

Pour ces travaux, la transformation retenue est celle qui permet de générer une ontologie à partir d'un modèle métier *UML/MOF*. Pour cela, un méta-modèle OWL (cf figure 4.6) conforme au *MOF* est en premier lieu défini. Ce dernier dérive du méta-modèle *RDFS* où les classes OWL sont des types de classes *RDFS*, tout comme les propriétés OWL sont des types de propriétés *RDF*. OWL permet cependant de définir une ontologie avec une sémantique plus riche :

- Définition de cardinalités sur les propriétés

- Définition de classes avec les opérateurs *union*, *intersection*, *complément*, etc.
- Définition d'individus représentant les ressources (i.e. instances de classe)

Pour effectuer un pont entre ces deux espaces technologiques, des liens sont définis entre les concepts issus de UML et ceux issus de OWL (cf tableau 4.5). Par exemple, une propriété UML définie comme un attribut associé à une classe est traduite par une propriété OWL indépendante de toute classe. La transformation de modèles prend en compte dans un premier temps ces correspondances, puis transforme les instances UML en individus afin de peupler la base de connaissances.

| UML         | OWL              |
|-------------|------------------|
| UML Class   | OWL Class        |
| Attribute   | DataTypeProperty |
| Association | ObjectProperty   |
| Instance    | Individual       |

TABLE 4.5 – Correspondances entre les concepts UML/OWL

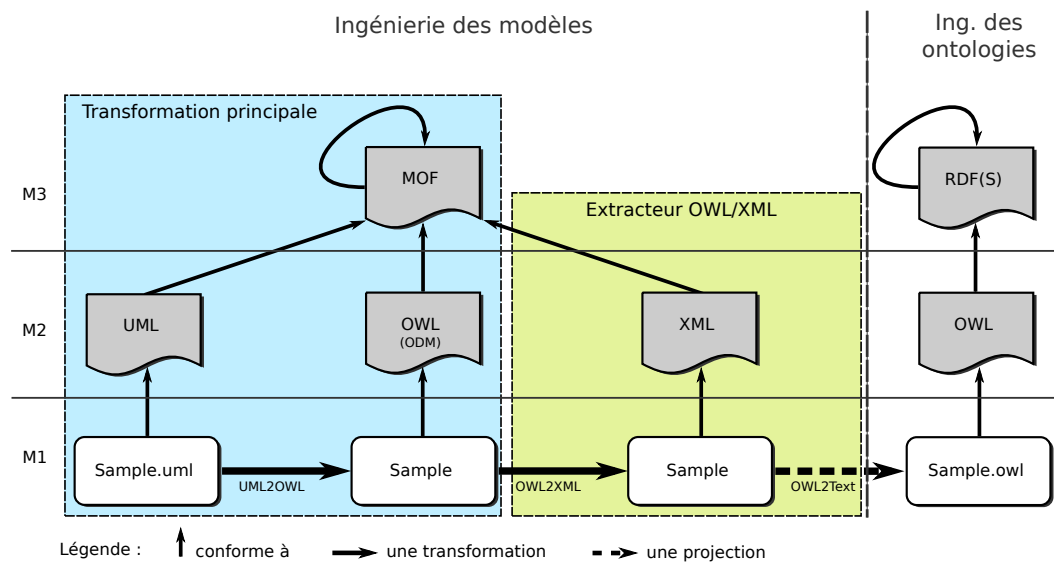


FIGURE 4.7 – Spécification ODM mise en oeuvre par transformations ATL

Une implémentation<sup>2</sup> de la spécification ODM utilisant le langage ATL est retenue dans ces travaux [Hillairet *et al.* 2008]. La figure 4.7 illustre l'implémentation du pro-

2. <http://www.eclipse.org/m2m/atl/usecases/ODMImplementation/>

cessus de génération de l'ontologie, composé de deux transformations :

1. **UML2OWL** prend en entrée un modèle UML et génère une ontologie conforme au méta-modèle *OWL* défini d'après la spécification *ODM*. Cette transformation se réalise en deux étapes : la première consiste à mettre en correspondance les concepts du modèle *UML* avec ceux des ontologies (les classes *UML* sont transformées en classes *OWL*, les *attributs* en *OWLDatatypeProperty*, etc. (cf tableau 4.5)). La deuxième étape de la transformation s'intéresse aux *instances* à convertir en *individuals* pour le peuplement de la base de connaissances.
2. **OWL2XML** permet ensuite de générer un document XML conformément à la syntaxe *OWL/XML*. Le document généré peut par la suite être chargé dans *Protégé* [Knublauch *et al.* 2004] pour manipuler l'ontologie.

### 4.3.2 Flot de conception pour la supervision

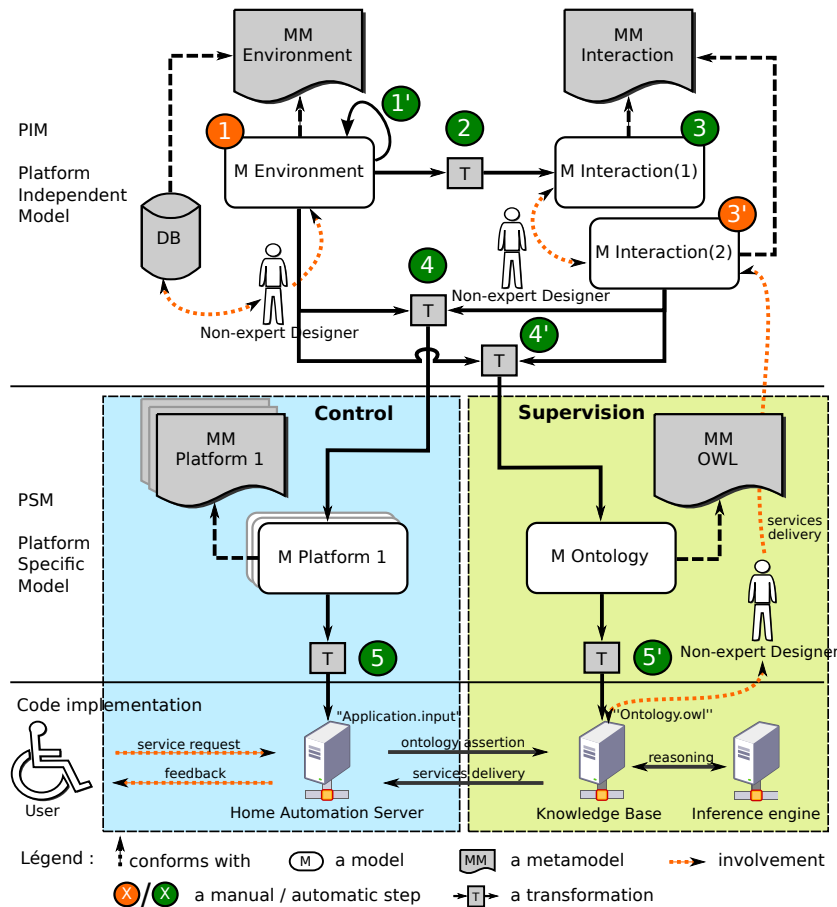


FIGURE 4.8 – Flot de conception pour le contrôle-commande domotique

La plupart des contributions sur la modélisation de contexte basée sur les ontologies ne considèrent pas la conception de ces dernières. Dans ces travaux, le peuplement de l'ontologie de contexte ne se fait pas de manière had-oc, mais est dérivé d'éléments issus de modèles métier dédiés à la conception d'un système domotique d'assistance. À partir du flot de conception dirigé par les modèles (cf chapitre 3) qui permet de générer la commande de ces systèmes, nous proposons de générer l'ontologie (*A-Box*) responsable de la supervision non-intrusive.

Les modèles métiers indépendants de toute plateforme servent de base pour le peuplement de l'ontologie. En effet, ils possèdent des informations essentielles sur l'environnement et les interactions définies pour un cas particulier (l'agencement de l'habitat, les équipements présents, leurs caractéristiques, mais aussi les interactions définies pour prendre en compte les besoins de l'utilisateur en situation de handicap). Lors de la transformation de modèles PIM → PSM (étape 4), certaines instances des modèles produits sont transformées en individus pour peupler l'ontologie (étape 5') sur laquelle il sera possible de raisonner à l'exécution.

### 4.3.3 Architecture de supervision

À partir des modèles définis à l'aide du *DSML*, la partie (*A-Box*) qui comprend les individus (i.e. les instances) de l'ontologie est générée. La partie terminologique (*T-Box*) est statique : elle représente les classes d'appartenance de ces individus. À chaque fois que l'on change de cas d'utilisation (e.g. agencement de l'habitat, redéfinition d'interactions), de nouveaux individus sont intégrés dans la *A-Box*. Associées à des règles SWRL, ces deux parties de l'ontologie représentent la base de connaissances sur laquelle repose l'application de supervision.

L'architecture logicielle présentée sur la figure 4.9 permet de mettre à jour l'ontologie à chaque fois que l'utilisateur demande l'activation d'un service domotique. Écrit en Java en utilisant l'API OWL [Horridge & Bechhofer 2011] pour interagir avec l'ontologie et le moteur d'inférences *HermiT* [Shearer *et al.* 2008], cette partie est basée sur six composants :

- **OWL Parser** prend en entrée les logs de commande provenant du serveur domotique pour les intégrer dans l'application
- **Supervision Unit** charge l'ontologie et ajoute les assertions associées aux logs (e.g. le log "*User1 Switch\_ON.TV1*" est traduit en assertions "*User1 isAskingForService Switch\_ON*" et "*User1 isUsing TV1*")
- **Inference Unit** fait appel au moteur d'inférences pour appliquer les règles SWRL sur l'ontologie composée des nouvelles assertions et envoie les inférences au *filtering unit*.
- **Filtering Unit** met à jour l'ontologie (e.g. mettre à jour la localisation de l'utilisateur consiste à supprimer les anciennes assertions et à filtrer les inférences pour n'ajouter que la localisation la plus précise possible)



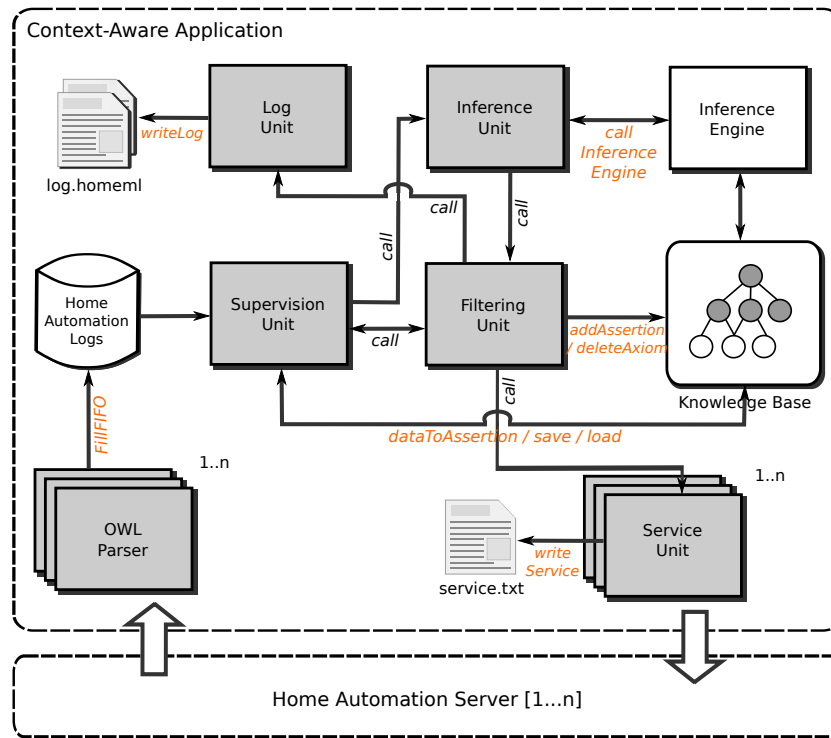


FIGURE 4.9 – Architecture logicielle pour la supervision basée sur les ontologies

- **Log Unit** sauvegarde les informations inférées sur la localisation et l'intention de l'utilisateur dans un fichier texte et sauvegarde les logs d'entrée conformément à la spécification *HomeML*<sup>3</sup> [Nugent *et al.*]
- **Service Unit** sauvegarde finalement les propositions de services de la même manière, avant de renvoyer une liste de services au serveur domotique

Le modèle de supervision non-intrusif s'inscrit donc dans une architecture logicielle pour son exécution. Il s'inscrit aussi dans un flot de conception dirigé par les modèles pour automatiser la génération des instances de l'ontologie (i.e. individus). Une évaluation de ce modèle basée sur les jeux de données du laboratoire Domus est présentée dans le chapitre 5.

3. *HomeML* est un format pour stocker des jeux de données avec la possibilité de les diffuser librement [McDonald *et al.*].

## **4.4 Conclusion du chapitre / contribution**

Dans ce chapitre, un modèle de supervision domotique non-intrusif a été présenté. Contrairement aux solutions existantes, cette supervision n'est pas basée sur l'utilisation de capteurs mais sur l'analyse sémantique des commandes domotiques envoyées par l'utilisateur. L'ontologie et les règles qui lui sont associées permettent à un moteur d'inférences d'extraire des informations habituellement captées. Suivant la manière dont l'utilisateur interagit avec son système domotique, sa localisation et son intention peuvent être déduites à l'aide d'inférences logiques, permettant d'aboutir à la proposition de services. Le chapitre suivant, qui présente les validations expérimentales des contributions de cette thèse, propose une validation de cette approche de supervision sur un cas d'étude réel.



# 5

## Validations expérimentales

*“L’expérience n’est qu’une observation  
invoquée dans un but de contrôle”*

Claude Bernard, 1813-1878

Les validations expérimentales menées dans ces travaux sont de deux ordres. Dans un premier temps, nous proposons de réaliser une série d’expérimentations pour évaluer le DSML et le flot de conception, qui correspondent respectivement aux chapitres 2 et 3 de ce mémoire. Ces expérimentations sont réalisées sur un cas d’étude réel : les appartements tremplins du centre de Kerpape (cf section 5.1). Dans un deuxième temps, nous proposons de valider le modèle de supervision non-intrusif correspondant au chapitre 4. Les appartements tremplins n’ayant pas encore reçu de patients, aucun jeu de données n’a pu être produit au moment de la rédaction de ce mémoire. Aussi, nous proposons de valider notre modèle de supervision sur les jeux de données tirés de l’habitat intelligent Domus de l’Université de Sherbrooke (cf section 5.2).

## Sommaire

---

|         |   |     |
|---------|---|-----|
| 5.1     | Évaluation du DSML et du flot de conception . . . . . | 125 |
| 5.1.1   | Cas d'étude : Kerpape Smart Home . . . . .            | 125 |
| 5.1.1.1 | Un tremplin vers l'indépendance . . . . .             | 125 |
| 5.1.1.2 | Description technique des appartements . . . . .      | 126 |
| 5.1.2   | Formalisation de l'évaluation . . . . .               | 129 |
| 5.1.2.1 | Critères d'évaluation . . . . .                       | 129 |
| 5.1.2.2 | Méthodes d'évaluation . . . . .                       | 129 |
| 5.1.2.3 | Hypothèses formulées / expérimentations contrôlées .  | 130 |
| 5.1.3   | Passage à l'échelle . . . . .                         | 131 |
| 5.1.3.1 | Utilisateur du cas d'étude . . . . .                  | 131 |
| 5.1.3.2 | Déroulement des étapes du flot . . . . .              | 132 |
| 5.1.4   | Expérimentation contrôlée / étudiants . . . . .       | 138 |
| 5.1.4.1 | Protocole d'expérimentation . . . . .                 | 138 |
| 5.1.4.2 | Résultats de l'expérimentation . . . . .              | 139 |
| 5.1.5   | Expérimentation contrôlée / ergothérapeutes . . . . . | 142 |
| 5.1.5.1 | Protocole d'expérimentation . . . . .                 | 142 |
| 5.1.5.2 | Résultats de l'expérimentation . . . . .              | 142 |
| 5.2     | Validation de la supervision . . . . .                | 143 |
| 5.2.1   | Cas d'étude : Domus Smart Home . . . . .              | 143 |
| 5.2.2   | Application pour le suivi de personnes . . . . .      | 145 |
| 5.2.3   | Résultats sur les jeux de données . . . . .           | 147 |
| 5.2.3.1 | Limiter les coûts . . . . .                           | 148 |
| 5.2.3.2 | Augmenter la sûreté de fonctionnement . . . . .       | 148 |

---

## 5.1 Évaluation du DSML et du flot de conception

L'évaluation du DSML et du flot de conception associé consiste en une série d'expérimentations basées sur un cas d'étude réel : les appartements tremplins du centre de Kerpape (cf section 5.1.1). La section 5.1.2 présente tout d'abord les critères à évaluer et motive chacune des expérimentations réalisées. Dans la première expérimentation, nous cherchons à appliquer toutes les étapes du flot de conception sur le cas d'étude pour aboutir à un système domotique à échelle réelle (cf section 5.1.3). Nous réalisons ensuite une série d'expérimentations pour intégrer des concepteurs non-experts en domotique en leur proposant de modéliser le système associé au cas d'étude. Une première expérimentation contrôlée est réalisée avec des étudiants (cf section 5.1.4). Enfin, nous proposons de reprendre cette expérimentation contrôlée avec des ergothérapeutes du centre de Kerpape pour s'approcher des conditions réelles dans lesquelles ce logiciel pourra être utilisé (cf section 5.1.5).

### 5.1.1 Cas d'étude : Kerpape Smart Home

#### 5.1.1.1 Un tremplin vers l'indépendance

Les appartements tremplins du CMRRF<sup>1</sup> de Kerpape (cf figure 5.1) ont été conçus pour évaluer l'autonomie des patients du centre en habitat individuel. L'objectif principal est d'anticiper le retour à domicile après hospitalisation et les travaux qui l'accompagnent. Ils représentent un cadre idéal pour que les patients, accompagnés d'ergothérapeutes, testent en situation réelle (dans le cadre d'un court séjour) les adaptations possibles, tant au niveau de l'aménagement architectural du lieu de vie que des solutions domotiques.

Parallèlement, les appartements tremplins servent de "*vitrine*" pour la formation d'artisans (e.g. plombiers, électriciens). Les artisans de la CAPEB<sup>2</sup> se voient proposer des journées de sensibilisation permettant de mieux comprendre le monde du handicap et de connaître les solutions possibles à apporter. Ces appartements servent enfin de support pour des travaux de recherche. Entièrement équipés en domotique, ils disposent de nombreux équipements (lumières, fenêtres, portes, volets, lits, etc.) que des personnes à mobilité réduite peuvent commander à partir de leur interface de commande (e.g. téléthèse) embarquée dans leur fauteuil. Les appartements tremplins disposent donc d'une infrastructure domotique complète permettant de tester de nouvelles technologies domotiques en situation réelle.

---

1. Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles de Kerpape : <http://www.kerpape.mutualite56.fr/>

2. CAPEB : Confédération de l'Artisanat et des Petites Entreprises du Bâtiment (<http://www.capeb.fr/>)



FIGURE 5.1 – Photos des appartements tremplins du CMRRF de Kerpape (chambre de l'appartement à gauche, cuisine du studio à droite)

### 5.1.1.2 Description technique des appartements

Les appartements tremplins sont composés d'un studio de 35m<sup>2</sup> et d'un appartement T2 de 75m<sup>2</sup> (cf figure 5.2). Le studio est équipé d'un bus domotique propriétaire *Legrand* et possède une cuisine, une chambre et une salle de bain. L'appartement est quant à lui équipé d'un bus domotique *KNX/EIB* ouvert qui répond à une norme de standardisation européenne. Ce dernier est composé de cinq pièces : une cuisine, un salon, une chambre, une salle de bain et une pièce de rangement<sup>3</sup>. Dans le local technique situé dans le hall d'entrée, un module IP, un serveur avec le logiciel *Domovea* pour la supervision de l'appartement équipé en *KNX* et des cartes d'acquisition audio/vidéo (portier) sont installés.

Les appartements tremplins sont équipés des types d'effecteurs (capteurs et actionneurs) suivants :

- **L** **Lumières** de deux types : les lampes de plafond et les appliques murales qui peuvent avoir des commandes de type *ON/OFF* et/ou *Variateur*
- **V** **Volets** roulants électriques avec les commandes *Monter/Descendre* et une indication du pourcentage d'ouverture
- **P** **Portes** équipées d'ouvre-portes électriques de deux types : battants et coulissants avec les commandes *Ouvrir/Fermer*
- **L** **Lits** électriques composés de deux sommiers individuels avec les commandes infrarouges *Monter/Descendre* pour chaque élément *Tête, Central* et *Pieds*
- **R** **Rails** au plafond permettant de se déplacer du lit vers la salle de bain et vice versa, avec les commandes *Monter/Descendre* et *Avancer/Reculer* en infrarouge
- **F** **Fenêtres** motorisées avec les commandes infrarouges *Ouvrir/Fermer* (système

3. La pièce de rangement n'étant pas accessible aux utilisateurs, elle n'est pas considérée dans ce cas d'étude.

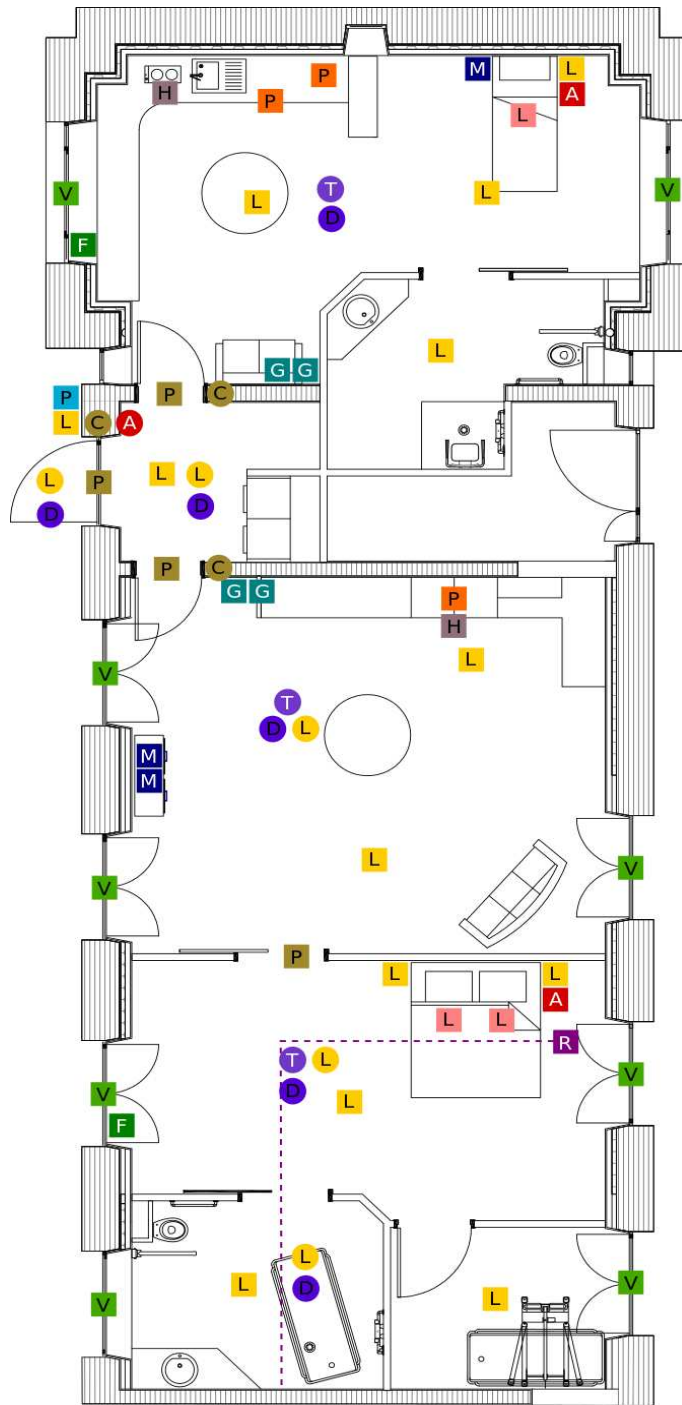


FIGURE 5.2 – Plan des appartements tremplins du CMRRF de Kerpape





| Capteurs    | Studio  |         |     | Appartement |       |         |     | Hall | Total |
|-------------|---------|---------|-----|-------------|-------|---------|-----|------|-------|
|             | Cuisine | Chambre | SDB | Cuisine     | Salon | Chambre | SDB |      |       |
| Contacteurs | 0       | 0       | 0   | 0           | 0     |         | 0   | 3    | 3     |
| Température | 1       | 0       | 0   | 1           | 0     | 1       | 0   | 0    | 3     |
| Luminosité  | 0       | 0       | 0   | 1           | 0     | 1       | 1   | 2    | 5     |
| Présence    | 1       | 0       | 0   | 1           | 0     | 1       | 1   | 2    | 6     |
| Anomalie    | 0       | 0       | 0   | 0           | 0     | 0       | 0   | 1    | 1     |

TABLE 5.1 – Nombre d’effecteurs (actionneurs et capteurs) selon l’emplacement

## 5.1.2 Formalisation de l’évaluation

### 5.1.2.1 Critères d’évaluation

Trois critères sont retenus pour évaluer le DSML et le flot de conception : l’expressivité, l’efficacité et l’adéquation :

**Expressivité** *“Le DSML est-t-il suffisant pour répondre à un ensemble satisfaisant de besoins utilisateurs ?”* Le pouvoir d’expression d’un langage de modélisation est difficilement mesurable. Évaluer son expressivité revient à déterminer dans quelle mesure le langage permet d’exprimer, de la manière la plus claire et la plus précise possible les concepts essentiels du domaine d’application auquel il s’adresse.

**Efficacité** *“L’utilisation du DSML apporte-t-elle des gains significatifs et quantifiables ? (e.g. productivité, qualité)”* L’utilisation du DSML doit permettre de définir de manière efficace les différents éléments essentiels pour la définition d’un environnement domotique et des besoins utilisateurs. L’utilisation d’outils de configuration existants par une personne non-avertie est à l’heure actuelle laborieuse, voire impossible. L’évaluation de l’efficacité concerne tant le langage de modélisation que le flot de conception.

**Adéquation** *“Le langage et le flot de conception sont-ils adaptés à leurs utilisateurs ?”* Les personnes non-expertes en domotique doivent être capables, à travers ce DSML, de configurer un système domotique. Les concepts à manipuler ainsi que leurs représentations graphiques doivent leur être accessibles. Dans notre contexte, il s’agit de l’accessibilité de l’outil de configuration pour les concepteurs non-experts en domotique.

### 5.1.2.2 Méthodes d’évaluation

L’évaluation des critères retenus passe par l’application de trois types de méthodes empiriques : l’étude de cas, l’expérimentation contrôlée et les enquêtes/sondages :

**Étude de cas** L'étude de cas permet d'évaluer l'expressivité du langage. La création du DSML a été motivée par le constat que les outils et langages existants dans le domaine de la domotique n'étaient pas suffisamment riches pour répondre pleinement aux besoins d'utilisateurs dépendants. Nous avons pour cela tenu à intégrer des concepts permettant de définir des interdictions sur un environnement domotique et de garantir la sûreté de fonctionnement (cf section 2.3.2 du chapitre 2).

**Expérimentations contrôlées** Les expérimentations contrôlées menées dans ces travaux permettent d'évaluer l'efficacité et l'adéquation (i.e. accessibilité) du DSML et du flot de conception associé. L'hypothèse que nous formulons, fil conducteur de ces travaux, est qu'une personne non-experte en informatique/électronique puisse intervenir dans le processus de conception d'un système domotique. Une évaluation réelle auprès de la population cible semble alors indispensable pour valider ce postulat.

**Enquête/Sondage** L'enquête, le sondage permet également d'évaluer l'efficacité et l'adéquation (i.e. accessibilité) du DSML et du flot de conception associé, en recueillant des informations subjectives liées au ressenti des participants.

Notre première expérimentation est une étude de cas : pour juger de l'expressivité du DSML, nous nous intéressons aux appartements tremplins sur lequel les étapes du flot sont déroulées. Ce cas d'étude est considérée suffisamment riche et réaliste pour apprécier le pouvoir d'expression du langage. Les deux autres expérimentations sont des expérimentations contrôlées auprès de concepteurs non-experts suivies d'une enquête/sondage pour évaluer l'efficacité et l'adéquation du DSML et du flot de conception. Nous proposons tout d'abord de mener une expérimentation contrôlée auprès d'étudiants, chargés de modéliser les appartements tremplins. Pour se rapprocher des conditions réelles d'utilisation, nous proposons de reproduire cette expérimentation auprès d'ergothérapeutes du centre de Kerpape. À la fin de chaque expérimentation, un questionnaire est distribué aux participants pour recueillir leurs impressions et les améliorations qui selon eux pourraient être apportées.

### 5.1.2.3 Hypothèses formulées / expérimentations contrôlées

L'hypothèse générale que nous formulons est que le logiciel soit suffisamment simple pour être accessible à des concepteurs non-experts et qu'il permette de répondre à un nombre de besoins satisfaisant. Pour vérifier cette hypothèse, nous nous intéressons à deux variables  $T$  (Temps) et  $C$  (Degré de conformité), qui sont relevées par la suite lors des expérimentations contrôlées. Elles se déclinent selon les trois activités de modélisation  $ENV$  (modélisation de l'environnement domotique),  $INT_1$  (modélisation des interactions de l'utilisateur 1) et  $INT_2$  (modélisation des interactions de l'utilisateur 2). La variable  $T$  se décline en  $T_{ENV}$ ,  $T_{INT_1}$  et  $T_{INT_2}$  et permet d'indiquer le temps (en minute) passé pour réaliser les différentes modélisations. La variable  $C$  se décline également en  $C_{ENV}$ ,  $C_{INT_1}$  et  $C_{INT_2}$  : elle représente un ratio de conformité dans l'intervalle

[0; 1] qui donne une mesure de la conformité des modèles produits par rapport aux spécifications.

|                    |                                    |        |                            |
|--------------------|------------------------------------|--------|----------------------------|
| Hypothèses $EXP_1$ | $H_1 : C_{ENV}$                    | $\geq$ | 0.8                        |
|                    | $H_2 : C_{INT}$                    | $\geq$ | 0.8                        |
|                    | $H_3 : T_{INT_{interaction\ n+1}}$ | $\leq$ | $T_{INT_{interaction\ n}}$ |
| Hypothèses $EXP_2$ | $H_1 : C_{ENV}$                    | $=$    | 1                          |
|                    | $H_2 : C_{INT}$                    | $=$    | 1                          |
|                    | $H_3 : T_{INT_{interaction\ n+1}}$ | $\leq$ | $T_{INT_{interaction\ n}}$ |

TABLE 5.2 – Hypothèses testées au cours des expérimentations  $EXP_1$  et  $EXP_2$

Le tableau 5.2 décline l’hypothèse générale en trois hypothèses  $H_1$ ,  $H_2$  et  $H_3$  que nous formulons avant l’expérimentation contrôlée auprès d’étudiants ( $EXP_1$ ) et avant celle menée auprès d’ergothérapeutes ( $EXP_2$ ). Pour  $EXP_1$ , nous considérons qu’un taux de conformité de 80% est satisfaisant pour  $ENV$  et  $INT$  (cf hypothèses  $H_1 : C_{ENV} \geq 0.8$  et  $H_2 : C_{INT} \geq 0.8$ ). En effet, nous nous intéressons dans cette expérimentation à la capacité des étudiants à traduire la description textuelle du tutoriel en modèles à l’aide du DSML : le résultat attendu correspond pour les étudiants à une mention très bien. Pour  $EXP_2$ , nous visons un taux de conformité de 100% (cf hypothèses  $H_1 : C_{ENV} = 1$  et  $H_2 : C_{INT} = 1$ ) avec des ergothérapeutes pour s’approcher le plus possible de la situation réelle : dans un contexte professionnel, le système doit être configuré sans aucune erreur. Enfin, nous cherchons à démontrer que l’utilisation du DSML est intuitive ( $H_3 : \text{gain en temps pour la modélisation successive d’interactions } INT_{(interaction)}$ ). La section suivante présente un passage à échelle dans les appartements tremplins qui ont servi de cas d’étude pour ces deux expérimentations.

### 5.1.3 Passage à l’échelle

#### 5.1.3.1 Utilisateur du cas d’étude

Pour le passage à l’échelle, nous considérons un utilisateur fictif<sup>4</sup> qui occupe l’appartement équipé d’un bus domotique KNX [KNX 2003]. Il s’agit d’une personne devenue hémiplegique suite à un AVC<sup>5</sup>. Il présente des capacités motrices réduites (difficultés pour la marche et membre supérieur droit non fonctionnel) nécessitant l’assistance d’une tierce personne (soignant, aide à domicile, membre de la famille, etc) pour réaliser certaines tâches de la vie courante. Se déplaçant en fauteuil roulant, sa mobilité réduite ne lui permet de marcher que pour réaliser les transferts lit/fauteuil et vice

4. Les appartements tremplins n’étant pas en fonction à l’heure actuelle, nous avons défini un utilisateur fictif dont le profil peut correspondre malgré tout à ceux des patients de Kerpape.

5. 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 & Wong 2009].

versa. La domotique représente une solution pertinente pour compenser cette situation en facilitant/automatisant le fonctionnement de son habitat. L'utilisateur présente également des troubles cognitifs débutants dans le cadre d'une démence de type Alzheimer<sup>6</sup> diagnostiquée avant l'AVC. Pour l'instant, il s'agit surtout de troubles mnésiques légers (i.e. petits oublis). Progressivement, ces troubles de mémoire vont s'accroître, 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 à ces besoins émergents. L'utilisateur a d'abord été hospitalisé dans une Unité Neuro-Vasculaire avant d'effectuer plusieurs mois de rééducation au centre de Kerpape. Pour préparer le retour au domicile (définition et dimensionnement des aides techniques et humaines), l'appartement de 75m<sup>2</sup> est mis à disposition. Il sera chaque fois que les besoins évoluent et nécessitent une nouvelle prise en charge du service de réadaptation et d'insertion sociale et professionnelle.

### 5.1.3.2 Déroulement des étapes du flot

**Importation du fichier de configuration** Un fichier de configuration ("*knx\_appartements.csv*") a été défini par les domoticiens lors de l'installation et l'intégration du bus domotique KNX dans l'appartement. Il permet, entre autres, de lister les adresses de groupe des équipements présents sur le bus. Un deuxième fichier de configuration ("*lircd\_appartements.conf*") reprend les commandes infrarouges des équipements.

| "knx_appartements.csv"   | "lircd_appartements.conf"   |
|--|---|
| <pre>"Group name"; "Address"  Entree centre;1/1/0 Salon centre var;1/1/1 Salon plan de travail;1/1/2 Salon cuisine;1/1/3 Chambre centre var;1/1/4 Chambre applique var;1/1/5 Rangement;1/1/6 WC;1/1/7 Exterieur;1/1/8 Variation Salon centre;1/1/9 Variation Chambre centre;1/1/10 Variation Chambre applique;1/1/11 ...</pre> | <pre>begin remote  name GEWA2 flags RAW_CODES CONST_LENGTH eps 30 aeps 100  gap 113790  begin raw_codes  name A 938 789 1770 1728 938 789 938 810 917 789 938 810 1834 1728 1770 1749 1770 874 874 ... end raw_codes ... end remote</pre> |

FIGURE 5.3 – Étape 0 : Importation du fichier de configuration

6. 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]. Des études récentes montrent un lien entre cette démence et les AVC [Nagai *et al.* 2009].

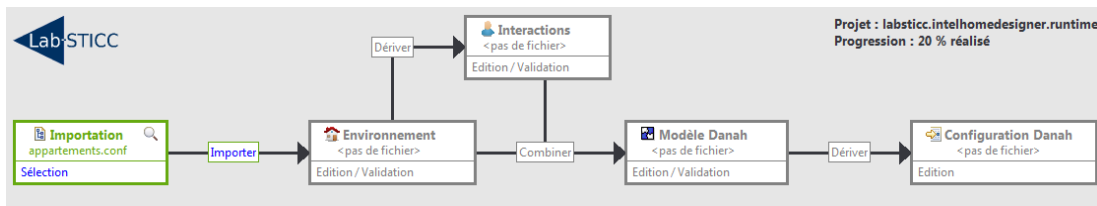


FIGURE 5.4 – Étape 0 : Importation à partir du tableau de bord

À partir du tableau de bord (cf figure 5.4), il est possible d’importer le fichier “*appartements.conf*” dans lequel sont précisés les chemins absolus vers ces deux fichiers de configuration (cf figure 5.3). Lors de l’appui sur le lien *Importer* du tableau de bord, les commandes (en bleu), les adresses (en vert) et les labels associés (en gris) sont intégrés dans le modèle d’environnement : un élément *IO Platform* est créé pour chaque fichier.

**Modélisation de l’environnement** La modélisation de l’environnement domotique représentée dans cette section considère l’appartement qu’occupe notre utilisateur. La figure 5.5 montre le modèle défini comprenant 4 composants agrégés à l’échelle d’une pièce (i.e. *Cuisine*, *Salon*, *Chambre* et *SDB*) et 18 composants élémentaires (i.e. équipements domotiques). Le *Hall d’entrée* a également été modélisé pour intégrer le bus domotique permettant le contrôle des portes, des volets, des fenêtres et des lumières. Une borne domotique (*Server 1*) est placée dans le local technique pour contrôler ce bus domotique à partir de l’application domotique Danah. Une deuxième borne (*Server 2*) est placée dans le salon pour le contrôle des équipements multimédia en infrarouge (i.e. télévision *TV1* et téléphone *PH1*).

**Transformation ENV → INT** Les services proposés à l’utilisateur, quels que soient leurs niveaux sémantiques et leurs types d’activation, sont décrits à l’aide d’opérations. De même, les modes de restriction sont décrits en termes d’états atteignables ou pas.

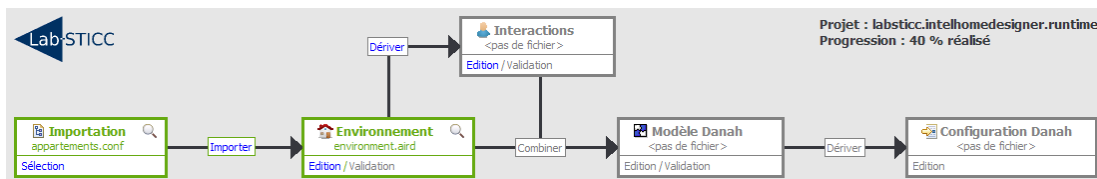


FIGURE 5.6 – Étape 2 : Transformation ENV→INT à partir du tableau de bord

La transformation de modèles “*Env2Int*” exploite les correspondances entre opéra-

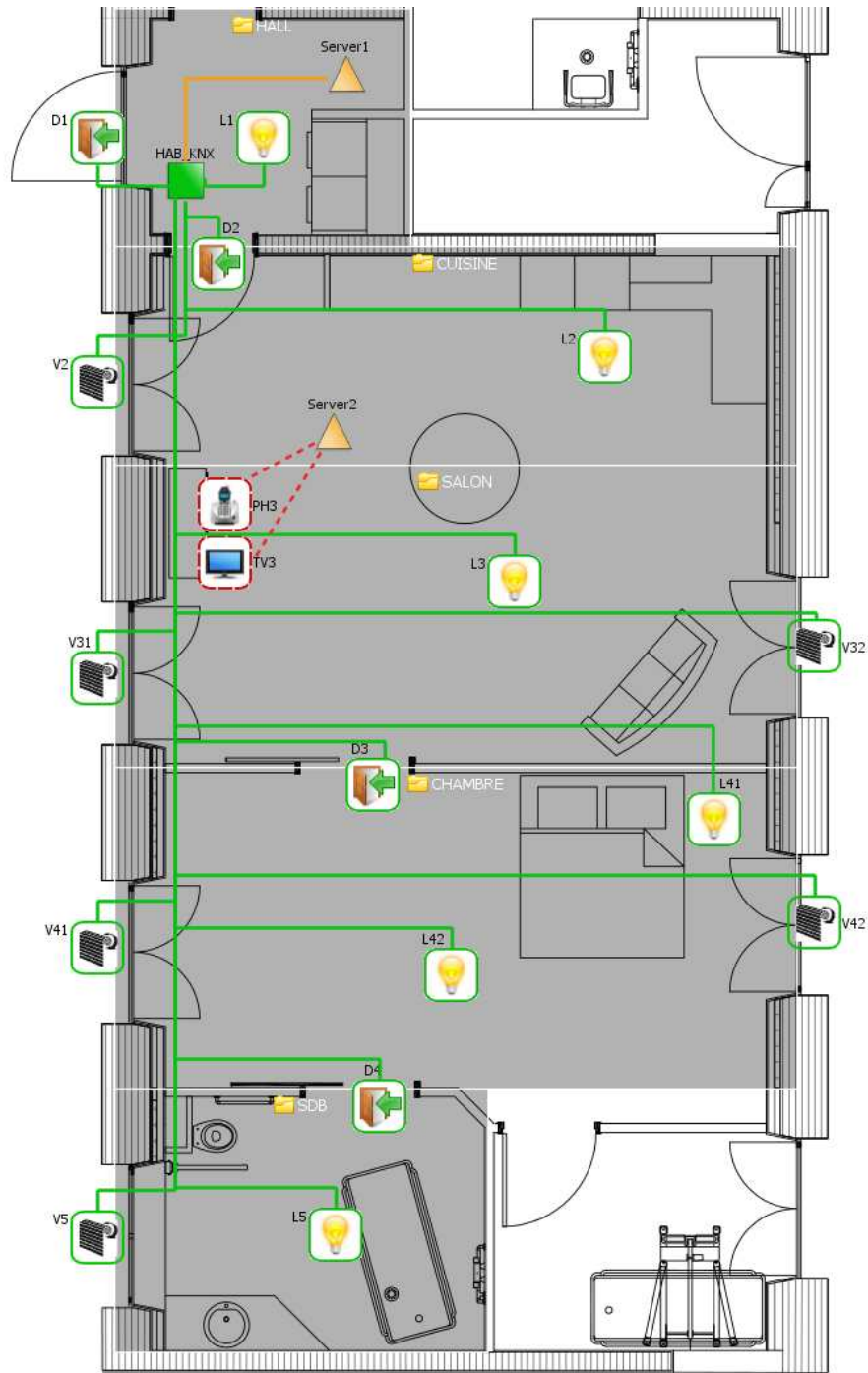


FIGURE 5.5 – Étape 1 : Modélisation de l'environnement (appartement T2)

tions-services et états-modes pour passer d'un modèle décrivant l'architecture du système vers une représentation de services propres à l'utilisateur. La figure 5.6 montre le tableau de bord avec le lien "Dérivier" qui permet de lancer cette transformation. Par exemple, la lumière *L1* et ses opérations *Allumer\_Lumière* et *Éteindre\_Lumière* se voient traduits par deux services *Allumer\_Lumière\_L1* et *Éteindre\_Lumière\_L1* activables manuellement. Les états associés à chacune de ces opérations sont ajoutés dans une liste du modèle d'interactions. Cela permet d'aboutir à un modèle d'interactions minimaliste comprenant un certain nombre d'états (qu'il sera par la suite possible d'interdire) et autant de services élémentaires que le concepteur pourra enrichir.

**Modélisation des interactions** Cette étape propose d'enrichir le modèle d'interactions généré pour l'adapter aux besoins de l'utilisateur. À partir des services élémentaires présents, il est possible de créer des scénarios et de les soumettre à des conditions d'activation automatique ou avec demande de confirmation. De la même manière, les modes d'interdiction peuvent être définis et soumis à des conditions particulières.

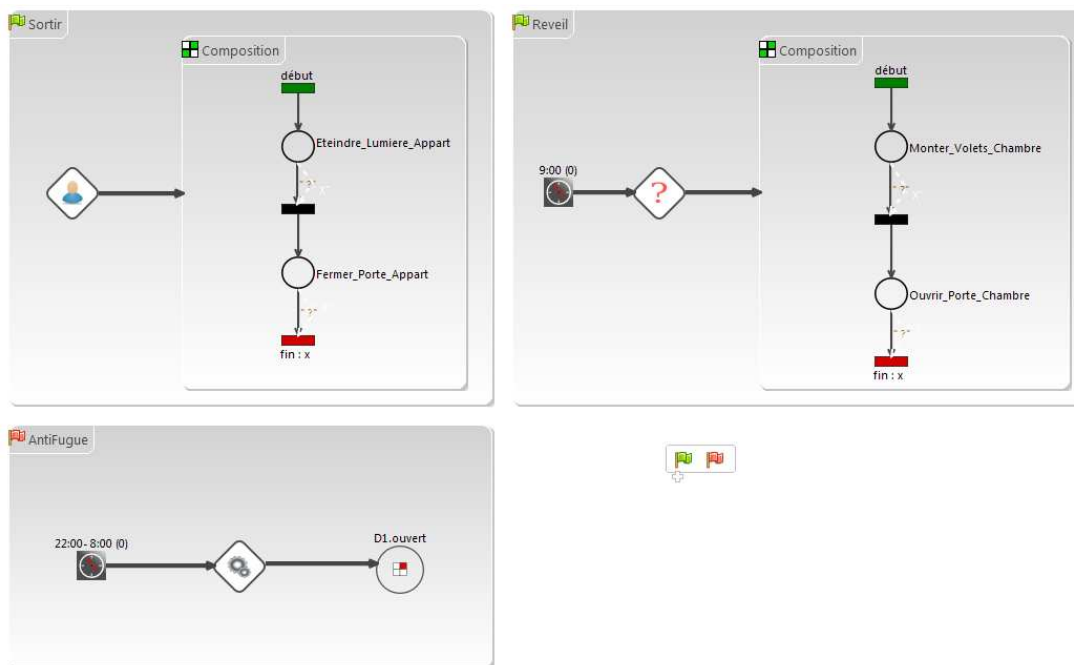


FIGURE 5.7 – Étape 3 : modélisation des interactions pour l'utilisateur du cas d'étude

Dans un premier temps, l'utilisateur a besoin d'automatisation pour compenser son handicap physique. Le scénario d'activation manuelle 5.1 (cf figure 5.7, en haut à gauche) permet d'éteindre toutes les lumières de l'appartement et de fermer les portes sous une seule activation :



$$\text{Sortir} : \text{UtilisateurVeut}(\text{EteindreLumiere\_Appart et FermerPorte\_Appart}) \quad (5.1)$$

Afin d'automatiser d'avantage le fonctionnement de l'appartement, le concepteur soumet des services (élémentaires ou scénarios) à des conditions d'activation automatique. Le service 5.2 (cf figure 5.7, en haut à droite) permet de descendre les volets de la chambre et d'allumer la lumières automatiquement lorsqu'il est 22h :

$$\begin{aligned} \text{SoirChambre} : \\ \text{Heure} = 22h \Rightarrow \text{SystExecute}(\text{AllumerLumiere\_L41 et DescendreVolet\_Chambre}) \end{aligned} \quad (5.2)$$

Quelques mois après son retour à domicile, l'utilisateur a besoin de revenir dans les appartements tremplins pour redéfinir des adaptations domotiques en réponse aux besoins qui ont évolués. 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, des services qui nécessitent une demande de confirmation de la part de l'utilisateur sont définis. Ces services, rythmant la journée, l'aident à se repérer dans le temps et stimulent sa mémoire à des moment clés de la journée (e.g. au réveil, au coucher, aux repas). Le service 5.3 demande à l'utilisateur s'il veut remonter les volets et ouvrir les portes de sa chambre à 9h :

$$\begin{aligned} \text{Reveil} : \\ \text{Heure} = 9h \Rightarrow \text{SystDemande}(\text{MonterVolet\_Chambre et OuvrirPorte\_Chambre}) \end{aligned} \quad (5.3)$$

La démence de type Alzheimer est à présent au stade sévère. De manière complémentaire aux services et en accord avec l'équipe médicale, un certain nombre d'interdictions sont définies dans le but de prévenir une éventuelle fugue et d'assurer la sécurité de l'utilisateur. Un mode Antifugue (interaction 5.4, en bas de la figure 5.7) est alors défini, interdisant l'ouverture de la porte d'entrée pendant la nuit :

$$\text{AntiFugue} : 22h < \text{Heure} < 8h \Rightarrow \text{SystInterdit}(D1.Ouverte) \quad (5.4)$$

À ce stade, l'automatisation des services qui étaient précédemment activables manuellement (ou avec demande de confirmation) va être privilégiée, permettant ainsi de retarder l'échéance du placement en institution.

**Transformation PIM → PSM** À partir des modèles indépendants de toute plateforme définis dans les étapes 1 et 3 du flot, il est maintenant possible de générer un modèle spécifique à la plateforme domotique Danah. La transformation "Intel2Danah" permet de générer le PSM Danah à partir des deux modèles définis à l'aide du DSML. La figure 5.8 montre le tableau de bord avec le lien "Combiner" qui permet de lancer cette transformation. À ce niveau, le concepteur fait le lien entre i) les labels issus des fichiers de configuration importés et les équipements domotiques modélisés et ii) les commandes issues des fichiers de configuration importés et les opérations des équipements

modélisés. À partir du modèle d'environnement, du modèle d'interactions et de ces correspondances, les *Units* et les *Actions* peuvent être générés.

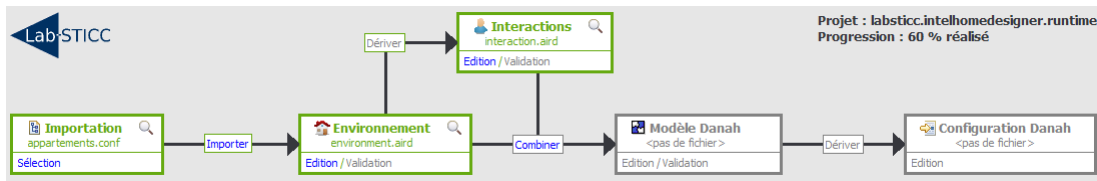


FIGURE 5.8 – Étape 4 : Génération du PSM Danah à partir du tableau de bord

**Transformation PSM → Code** À partir du PSM issu de l'étape précédente, il est possible de générer un fichier de configuration pour la plateforme domotique Danah. Un générateur de code *Acceleo* traduit chaque élément du PSM Danah en texte. Une fois le fichier de configuration généré, ce dernier est embarqué dans une borne domotique supportant la plateforme Danah. L'utilisateur peut ainsi s'y connecter pour solliciter les interactions précédemment définies. La figure 5.9 donne un aperçu de l'interface graphique présente sur le PDA de l'utilisateur.



FIGURE 5.9 – Interface utilisateur Danah affichée sur le PDA

À la fin de cette expérimentation, nous obtenons les résultats attendus, à savoir une implémentation dans les appartements tremplins de Kerpage. Les interactions définies dans les étapes de modélisation peuvent ainsi être sollicitées.

## 5.1.4 Expérimentation contrôlée / étudiants

### 5.1.4.1 Protocole d'expérimentation

L'expérimentation contrôlée, au coeur de l'évaluation de nos travaux, se base sur la méthode *GQM* (*Goal - Question - Metric*) introduite dans [Basili & Weiss 1984]. D'après la méthode *GQM*, une expérimentation doit être définie selon cinq critères : l'objet de l'étude, le but recherché, les caractéristiques observées, la perspective et le contexte de l'étude. Tous ces points sont repris et appliqués pour formaliser notre expérimentation :

1. **Objet de l'étude** : modéliser un environnement domotique (agencement, configuration des équipements), puis définir les interactions utilisateur-système pour répondre au cas d'étude déployé dans la section précédente. À partir d'une spécification textuelle (qui pourrait être formulée par un utilisateur), l'objectif est d'aboutir à une modélisation complète de l'appartement tremplin à l'aide de l'outil de configuration.
2. **But recherché** : évaluer l'efficacité de personnes non-expertes en domotique à réaliser les tâches objets de cette étude mais aussi évaluer l'adéquation du langage et de la plateforme associée. Pour l'adéquation de l'outil et des concepts manipulés, une enquête portant sur le degré d'utilisabilité de l'outil est menée. L'efficacité est évaluée en fonction de mesures effectuées sur i) les temps de modélisation et ii) la conformité des modèles produits par rapport à une solution cible attendue (conforme aux spécifications fournies).
3. **Caractéristiques observées** : l'efficacité est mesurée par le temps passé pour réaliser l'activité de modélisation. La qualité est mesurée par le degré de conformité des modèles produits en fonction d'un modèle cible.
4. **Perspective** : cette expérimentation est menée pour conforter notre hypothèse selon laquelle une personne non-experte en domotique peut être partie prenante de la conception d'un système domotique. La perspective qui découle de cette expérimentation est une mise en situation réelle avec des ergothérapeutes.
5. **Contexte de l'étude** : les sujets sont des étudiants du département *QLIO*<sup>7</sup>, dans lequel sont enseignés les qualités de communication, de travail en équipe et le sens de l'organisation. Nous considérons à ce titre que la population testée n'est pas experte en domotique et faisons l'hypothèse qu'elle a un niveau de compétences identique à la population cible qui sera amenée à utiliser le logiciel.

L'expérimentation se déroule sur une séance de TP de quatre heures. Chaque étudiant occupe un poste de travail avec un PC sur lequel le logiciel de configuration a préalablement été installé. Deux activités de modélisation sont ainsi réalisées : *ENV* fait référence à la modélisation de l'environnement domotique alors que *INT* représente la définition des interactions utilisateur-système. Les huit étudiants sont répartis en deux

---

7. *QLIO* : Qualité, Logistique Industrielle et Organisation (<http://www-iutlorient.univ-ubs.fr/>)

groupes homogènes *A* et *B*. Pour chaque groupe, l'activité de modélisation débute par la partie 1 (*ENV*) puis continue avec la partie 2 (*INT*<sub>1</sub> pour le groupe *A* ou *INT*<sub>2</sub> pour le groupe *B*). L'expérimentation s'est déroulée comme suit :

1. Présentation du contexte et des objectifs de l'expérimentation (présentation de la méthodologie puis distribution du formulaire de consentement éclairé)
2. Découverte du logiciel (déroulement du flot de conception sur un exemple)
3. Présentation du cas d'étude : cf annexe E, tutoriel 1
4. Activité de modélisation *ENV* (relevé du temps + enregistrement des modèles)
5. Activité de modélisation *INT* (relevé du temps + enregistrement des modèles)
6. Distribution de l'enquête/sondage et recueil des impressions

#### 5.1.4.2 Résultats de l'expérimentation

**Analyse quantitative (expérimentation contrôlée)** Les résultats de l'expérimentation sont analysés en fonction des variables *T* (temps) et *C* (conformité). Le temps est relevé après chaque activité de modélisation (*ENV* et *INT*). Les modèles produits sont également enregistrés pour être comparés aux modèles cibles. Pour relever le temps de modélisation de chaque interaction, un logiciel de capture vidéo a été utilisé.

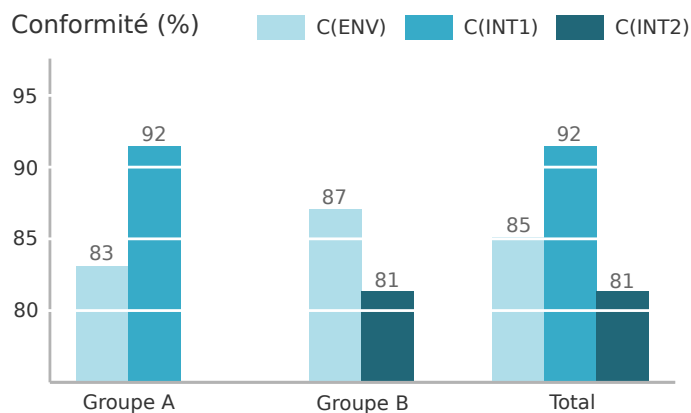
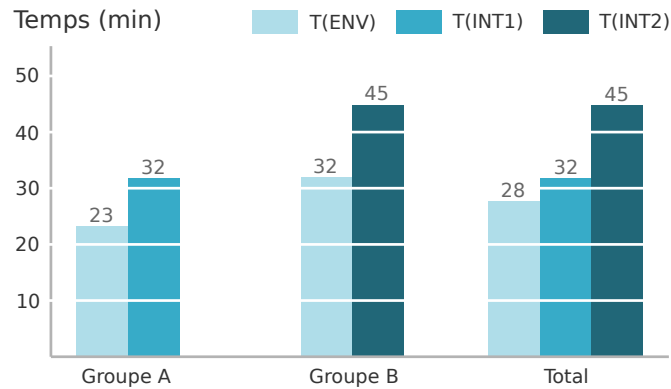


FIGURE 5.10 – Résultats de l'expérimentation - C : conformité (%)

Les figures 5.10 et 5.11 présentent, pour chaque groupe et pour chaque activité de modélisation, les relevés des temps et des taux de conformité des modèles produits. Par exemple, on peut y lire que les modèles produits (en 23 minutes) par le groupe *A* pour l'activité de modélisation *ENV* sont conformes à 83% aux modèles attendus<sup>8</sup>. Globalement, les activités de modélisation durent entre 23 et 45 minutes. Le taux de conformité sans vérification syntaxique est compris entre 81% et 92%. Ces chiffres traduisent

<sup>8</sup> Pour l'activité de modélisation *ENV*, 83% correspond à 46 propriétés sur 55 (attributs et liens d'association) correctement définies.

FIGURE 5.11 – Résultats de l'expérimentation -  $T$  : temps (min)

une bonne capacité de traduction de la description textuelle issue du tutoriel en modèles définis à l'aide du DSML. On peut toutefois constater une disparité entre les deux groupes : le groupe  $B$  met globalement plus de temps que le groupe  $A$ , avec un taux de conformité plus faible.

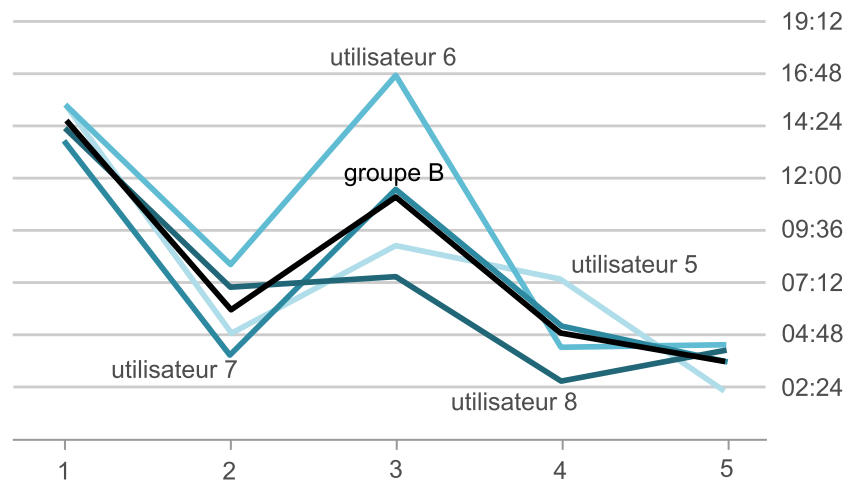


FIGURE 5.12 – Temps unitaire de modélisation d'interactions

La figure 5.12 montre le temps passé par le groupe  $B$  pour définir les cinq interactions de l'activité de modélisation  $INT_2$ . On remarque que la modélisation d'une interaction est bornée entre 2min30s et 17min. En écartant l'interaction 3, on constate que le temps de modélisation baisse au fur et à mesure des interactions. Le temps pour modéliser la première interaction est compris entre 13min43s et 15min24s alors que la dernière interaction a été modélisée en moins de 4min19s. Cette tendance reflète un

apprentissage à court terme des actions à réaliser pour définir des interactions (alors que ces dernières sont présentées par ordre croissant de complexité). Après discussion avec les étudiants, le temps important pris pour définir l'interaction 3 s'explique par une mauvaise compréhension textuelle de l'interaction à modéliser.

Les hypothèses  $H_1$ ,  $H_2$  et  $H_3$  de la section 5.1.2.3 ont toutes été vérifiées :

- $H_1$  : tous les sujets de l'étude, qui ne possèdent aucune compétences en domotique, ont été capables d'intégrer les équipements du cas d'étude et de configurer les bornes domotiques avec un taux de conformité de 85% en moyenne. Le temps mis pour cette partie est relativement encourageant : 28 min en moyenne.
- $H_2$  : tous les sujets de l'étude ont été capables de traduire les besoins utilisateur du cas d'étude spécifiés sous forme textuelle en modèles définis à l'aide du DSML avec un taux de conformité de 87% en moyenne en 39min.
- $H_3$  : le temps de modélisation pour chaque interaction du cas d'étude diminue au cours du temps ( $T_{INT_{interaction\ n+1}} \leq T_{INT_{interaction\ n}}$ ). Cette tendance permet de confirmer l'apprentissage à court terme / l'intuitivité du DSML et du logiciel.

**Analyse qualitative (enquête/sondage)** Le questionnaire distribué à la fin de l'expérimentation contrôlée (cf annexe C) permet de mieux comprendre les phénomènes observés qui ont été chiffrés dans la section précédente. Les étudiants qui ont participé à l'expérimentation sont sensibilisés à la domotique (seule une personne sur huit ne connaissait pas ce terme avant l'expérimentation), mais aucun d'entre eux n'avait auparavant utilisé de logiciel de configuration domotique. L'outil manipulé a été jugé peu complexe à utiliser (complexité : 1,1/4). La description textuelle a dans l'ensemble été la plus difficile à comprendre, cela essentiellement à cause de la présence de termes jugés "trop techniques" (difficulté : 1,3/4). Malgré tout, une fois cette étape passée, la traduction à l'aide de l'outil a été jugée abordable : 75% ont jugé la traduction avec l'outil plus facile que la compréhension textuelle.

L'activité de modélisation *INT* a globalement été préférée. Elle a pourtant été jugée sensiblement plus complexe que *ENV* (difficulté perçue : 0,8/4 pour *ENV* contre 1,1 pour *INT*). Ce résultat se vérifie pour les deux groupes (avec des modèles *INT* différents). La définition de scénarios (représentation graphique : *RdP*) a été jugée assez intuitive (intuitivité : 2,25/4). Les observations directes et les échanges avec les participants pendant l'expérimentation ont également permis de comprendre pourquoi l'interaction 3 a pris beaucoup plus de temps que l'interaction précédente pour le groupe *B* : la description textuelle de cette interaction n'a pas été correctement comprise. (Ce n'a pas été le cas pour le groupe *A* avec l'activité de modélisation *INT1*)

Finalement, une fois la description graphique comprise, la grande majorité trouve que l'outil est "accessible et pas forcément difficile à manipuler" : "l'assimilation des objets se fait assez rapidement dans l'ensemble". Les instructions fournies étant assez claires (com-

préhension : 1,75/4), on peut espérer une amélioration du degré de conformité des modèles produits en ajoutant du temps d'apprentissage. Un apprentissage plus important aurait ainsi permis de "*se familiariser avec les objets graphiques*" et de "*faciliter la prise en main de l'outil*".

## 5.1.5 Expérimentation contrôlée / ergothérapeutes

### 5.1.5.1 Protocole d'expérimentation

Pour confirmer les tendances qui se dégagent de la première expérimentation réalisée auprès d'étudiants, nous proposons de reproduire cette expérimentation avec des ergothérapeutes du centre de Kerpape. Deux ergothérapeutes ont ainsi accepté de participer à l'évaluation du DSML et du flot de conception sur un cas d'étude réduit, en utilisant le tableau de bord (cf section 3.3.1.2 du chapitre 3) développé pour automatiser les étapes de la conception. L'objectif de cette expérimentation est donc de générer un fichier de configuration fonctionnel (avec un taux de conformité de 100% pour les modèles à définir). L'expérimentation s'est déroulée sur deux séances de deux heures pour chaque ergothérapeute. Dans la première séance, l'utilisateur est guidé dans son activité de modélisation : tout en suivant le tutoriel (cf annexe E, tutoriel 2), il peut demander des informations et des conseils à tout moment. Une semaine après, l'utilisateur suit de nouveau le tutoriel mais de manière autonome (seul l'utilisateur 1 a pu participer à cette dernière séance).

### 5.1.5.2 Résultats de l'expérimentation

**Analyse quantitative (expérimentation contrôlée)** Les résultats de l'expérimentation sont analysés en fonction des variables dépendantes  $T$  (temps) et  $C$  (conformité) relevées lors de la deuxième séance (la première séance est considérée comme une phase d'apprentissage). L'utilisateur 1 a par exemple mis 9 minutes pour l'activité de modélisation  $ENV$  avec un taux de conformité de 100%. Les cas d'étude n'étant pas les mêmes, ce chiffre ne peut pas être comparé aux résultats trouvés lors de la première expérimentation. En revanche, le temps mis pour définir les interactions lors de l'activité de modélisation  $INT$  peut être comparé : il s'agit en effet des mêmes types d'interactions. Pour cette activité de modélisation, nous avons obtenu de meilleurs résultats qu'avec les étudiants. Cela est sûrement lié au fait que la même expérimentation est répétée deux fois en une semaine (apprentissage à long terme). L'utilisateur 1 a par exemple mis 16 minutes pour modéliser 3 interactions (cf annexe E, tutoriel 2) avec un taux de conformité de 100%.

Les hypothèses  $H_1$ ,  $H_2$  et  $H_3$  de la section 5.1.2.3 ont également pu être vérifiées pour cette expérimentation :

- $H_1$  : tous les sujets de l'étude ont été capables de modéliser l'environnement du cas d'étude avec un taux de conformité de 100% (en utilisant le tableau de bord).

- $H_2$  : tous les sujets de l'étude ont été capables de modéliser les interactions du cas d'étude avec un taux de conformité de 100% (en utilisant le tableau de bord).
- $H_3$  : tout comme dans l'expérimentation auprès d'étudiants, le temps de modélisation pour chaque interaction du cas d'étude diminue au cours du temps.

**Analyse qualitative (enquête/sondage)** Au vu du faible nombre de participants pour cette expérimentation, il nous semble d'autant plus important de recueillir les impressions et les retours des ergothérapeutes. Les résultats du questionnaire (cf annexe C) montrent que les utilisateurs ont déjà ressenti le besoin de configurer un système domotique dans le cadre de leur activité professionnelle. Suite à l'expérimentation du logiciel, ils pensent être capable de le faire avec ce dernier. Néanmoins, tout comme dans la première expérimentation, l'activité de modélisation *INT* a été jugé complexe à réaliser. Des difficultés ont notamment été relevées au niveau du "*vocabulaire utilisé*", de l'emplacement des éléments dans la palette ("*où aller chercher les liens*") et de la définition de scénarios.

Le cas d'étude n'étant pas le même dans les deux expérimentations contrôlées, il n'est malheureusement pas possible de comparer les temps de modélisation. À la suite de ces deux expérimentations, on peut malgré tout affirmer que la modélisation de l'environnement domotique (*ENV*) peut être accessible à une personne non-experte en domotique ( $C_{ENV} > 0.8$  pour les étudiants sur un cas d'étude complet, sans vérification syntaxique). Les concepteurs qui ont participé sont également capables d'utiliser le DSML pour définir des interactions adaptées aux besoins de l'utilisateur ( $C_{INT} > 0.8$  pour la première expérimentation). On peut cependant s'attendre à de meilleurs résultats avec un temps d'apprentissage plus important ainsi qu'une amélioration de l'interface et de la syntaxe graphique.

## 5.2 Validation de la supervision

Pour évaluer la supervision non-intrusive basé sur les ontologies, nous prenons le cas d'étude de l'appartement Domus de l'Université de Sherbrooke<sup>9</sup> (cf section 5.2.1). Nous considérons d'abord un extrait des jeux de données tirés de cet habitat pour illustrer notre approche pour une application de suivi de personnes (cf section 5.2.2). Les résultats sur l'ensemble des jeux de données sont présentés dans la section 5.2.3. Ils mettent en avant l'intérêt de notre proposition, tant pour limiter l'utilisation de capteurs que pour augmenter la sûreté de fonctionnement.

### 5.2.1 Cas d'étude : Domus Smart Home

L'appartement Domus (cf figure 5.13) comprend une chambre, une salle de bain, une cuisine, une salle à manger et un salon. Il est équipé de différents types de cap-

---

9. Université de Sherbrooke : [domus.usherbrooke.ca](http://domus.usherbrooke.ca)



teurs<sup>10</sup> : des détecteurs de mouvements infrarouges pour localiser l'utilisateur ; des capteurs électromagnétiques pour l'ouverture et la fermeture des portes ; un tapis avec capteurs de pression placé dans le hall d'entrée qui détecte l'entrée dans l'appartement ; des contacteurs pour les interrupteurs, les armoires et les tiroirs qui envoient un événement à chaque fois que l'utilisateur ouvre ou ferme ces équipements. Des débitmètres détectent enfin quand la chasse d'eau est tirée ou quand un robinet est ouvert/fermé.

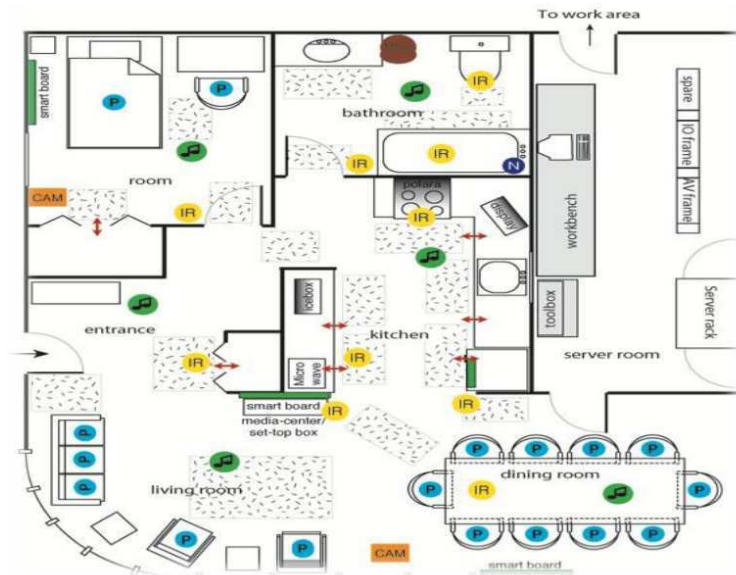


FIGURE 5.13 – Plan de l'appartement Domus tiré de [Domus ]

Pour la création des jeux de données [Domus ], six chercheurs du laboratoire Domus ont participé à deux expérimentations. Dans la première expérimentation (série 1), l'utilisateur effectue la routine du matin (se lever, faire sa toilette, préparer et prendre son petit-déjeuner) comme il a l'habitude de le faire chez lui. Dans la deuxième expérimentation (série 2), l'utilisateur répète cette routine avec l'introduction d'une contrainte qui dure au plus dix minutes. Dans les deux séries d'expérimentation qui durent environ 45 minutes, l'utilisateur peut utiliser tous les équipements disponibles dans l'appartement et décide de l'ordre des activités qui composent sa routine. Chaque donnée de capteurs est identifiée par son nom, sa valeur ou son état et une information temporelle qui indique quand le capteur a été déclenché. Le tableau 5.3 montre un exemple de logs enregistrés pendant les expérimentations :

10. Nous listons uniquement les capteurs présents dans les jeux de données Domus : les capteurs de son et les caméras intégrés dans l'appartement ne sont par exemple pas mentionnés.

| Time       | Sensor ID | Sensor name | Sensor location | State/Value |
|------------|-----------|-------------|-----------------|-------------|
| 09 :08 :18 | 5105      | Light       | Bedroom         | SwitchON    |
| 09 :08 :28 | 0215      | Door        | Bedroom         | Open        |
| 09 :08 :34 | 5101      | Light       | Kitchen         | SwitchON    |
| 09 :08 :38 | 5105      | Light       | Bedroom         | SwitchOFF   |
| 09 :08 :42 | 0215      | Door        | Bedroom         | Close       |

TABLE 5.3 – Exemple de logs issus des jeux de données Domus

### 5.2.2 Application pour le suivi de personnes

Pour illustrer notre approche, une application de suivi de personnes pour la proposition dynamique de services est proposée. L'extrait considéré (utilisateur 1, jour 2) est présenté dans le tableau 5.4. Pour mettre en avant l'intérêt d'analyser les requêtes de l'utilisateur, nous proposons de convertir les logs de capteurs issus de ces jeux de données en logs de commandes. Cette conversion n'est possible que pour les équipements qui sont potentiellement composés du couple {*capteur*, *actionneur*} (e.g. une porte peut être équipée d'un actionneur (ouvre-porte) pour la commande à distance et d'un capteur pour en vérifier l'état). Les logs d'équipements qui ne peuvent pas être activés à distance ne sont donc pas traduits. Prenons l'exemple du robinet de la cuisine : nous considérons que ce dernier ne peut pas être activé à distance à partir de l'interface de l'utilisateur. Ainsi, les logs du débitmètre monté sur ce robinet (i.e. *F101*) ne sont pas traduits en logs de commande : ils ne sont donc pas interprétés dans notre cas d'étude. Dans le tableau 5.4, les logs de commandes sont représentés par le symbole "°" alors que les logs de capteurs (qui n'ont pas pu être traduits) sont représentés par "\*". Nous dégageons ainsi cinq grandes étapes de cet extrait, permettant d'illustrer l'analyse sémantique des commandes pour inférer la localisation et l'intention de l'utilisateur et ainsi aboutir à la proposition contextuelle de services (cf figure 5.14).

**Étape 1** L'utilisateur *User 1* allume la lumière de la chambre "[Log 1] : *Light5105-SwitchON*" (cf tableau 5.4). Ce log est introduit dans l'ontologie à travers les assertions<sup>11</sup> : "[Assertion 1] : *User1 isAskingForService SwitchON*" et "[Assertion 2] : *User1 isUsing Light5105*". Sachant que l'ontologie définit *Light5105* comme une ressource d'activation locale placée dans la chambre *Room* et que cette dernière possède le service élémentaire *SwitchON* avec l'effet associé *Luminosity*, les règles suivantes peuvent être appliquées : "[Rule 1] : *isLocatedIn(LocalDevice, Room), isUsing(User, LocalDevice) → isLocatedIn(User, Room)*" et "[Rule 2] : *hasEffect(BasicService, Effect), isAskingForService(User, BasicService) → wants(User, Effect)*". L'interprétation de ces règles par le moteur d'inférences permet alors d'aboutir aux inférences "[Inference 1] : *User1 isLocatedIn Bedroom*"

11. Chaque log domotique est traduit avec les propriétés d'objets *isAskingForService* et *isUsing*.

| N° |   | Time    | ID   | Name        | Location    | Service/State |
|----|---|---------|------|-------------|-------------|---------------|
| 1  | * | 8:56:17 | 5105 | Light       | Bedroom     | SwitchON      |
| 2  | * | 8:56:26 | 0215 | Door        | Bedroom     | Open          |
| 3  | * | 8:56:31 | 5101 | Light       | Kitchen     | SwitchON      |
| 4  | * | 8:56:35 | 5105 | Light       | Bedroom     | SwitchOFF     |
| 5  | * | 8:56:39 | 0215 | Door        | Bedroom     | Close         |
| 6  | * | 8:56:44 | 0216 | Door        | Bathroom    | Open          |
| 7  | * | 8:56:46 | 5106 | Light       | Bathroom    | SwitchON      |
| 8  | ° | 8:56:51 | Fl01 | T.ColdWater | Bathroom    | Open state    |
| 9  | ° | 8:56:54 | Fl01 | T.ColdWater | Bathroom    | Close state   |
| 10 | ° | 8:57:09 | IR01 | Infrared    | KitchenSink | Close state   |
| 11 | ° | 8:57:12 | IR02 | Infrared    | KitchenOven | Close state   |
| 12 | * | 8:57:12 | 5106 | Light       | Bathroom    | SwitchOFF     |

\* : logs de commande / ° : logs de capteurs

TABLE 5.4 – Jeu de données de la série 1 / utilisateur 1 / jour 2 (extrait retenu)

et “[Inference 2] : User1 wants Luminosity”.

**Étape 2** L'utilisateur veut maintenant se rendre dans la salle de bain : après avoir ouvert la porte de la chambre “[Log 2] : Door0215.Open”, il allume la lumière de la cuisine : “[Log 3] : Light5101.SwitchON”. Comme la lumière est considérée comme une ressource d'activation locale, les inférences suivantes sont réalisées en utilisant les mêmes règles que celles de l'étape 1 : “[Inference 3] : User1 isLocatedIn Kitchen” et “[Inference 4] : User1 wants Luminosity”.

**Étape 3** Le système propose à l'utilisateur d'éteindre la lumière de la chambre. Cette proposition est basée sur la propriété d'effet inverse (e.g. *Luminosity isInverseOf Darkness*). L'utilisateur étant localisé dans la cuisine et voulant de la luminosité, le système lui propose d'éteindre les lumières dans les pièces où il ne se trouve pas. Dans cette application, ce service est accepté par l'utilisateur : “[Log 4] : Light5105.SwitchOFF”.

**Étape 4** Pour aller dans la salle de bain, il demande l'ouverture de la porte : “[Log 6] : Door0216.Open”. Cette porte étant liée à deux pièces avec la propriété *isBetween*, les règles suivantes peuvent être appliquées : “[Rule 3] : Door(?d), Room(?r), User(?u), isUsing(?u,?d), isUsingFrom(?d,?r) → isLeaving(?u,?r)” et “[Rule 4] : Door(?d), Room(?r1), Room(?r2), User(?u), isBetween(?d,?r1), isBetween(?d,?r2), isLeaving(?u,?r1), isLinkedWith(?r1,?r2) → isEntering(?u,?r2)”. L'interaction entre l'utilisateur et la porte de la salle de bain permet au moteur d'inférences d'extraire les connaissances suivantes : “[Inference 5] : User1 isEntering Bathroom” et “[Inference 6] : User1 isLeaving Kitchen”.

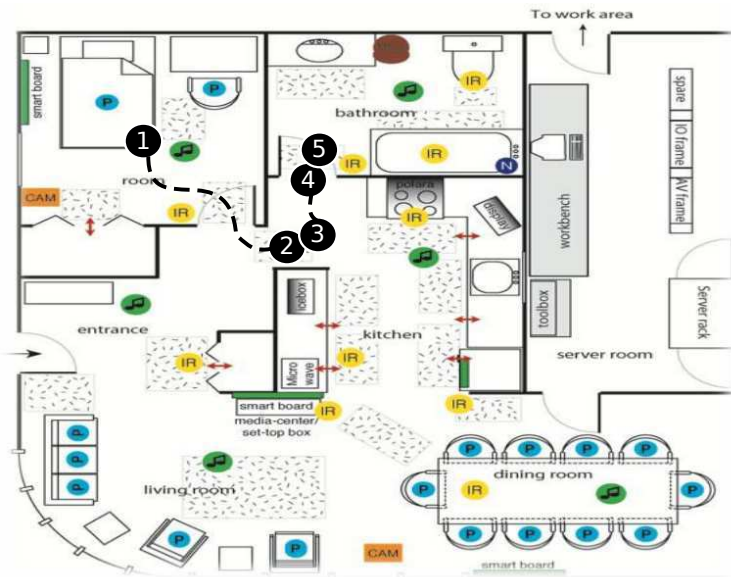


FIGURE 5.14 – Application pour le suivi de personnes dans l'appartement Domus

**Étape 5** Finalement, le système propose à l'utilisateur d'allumer la lumière de la salle de bain "[Log 7] : *Light5106.SwitchON*" à partir de son intention et de sa localisation précédemment inférées. Ces inférences sont rendues possible par les règles suivantes : "[Rule 5] : *BasicService(?b), Device(?d), Effect(?e), Room(?r), System(?s), hasToProvide(?s,?e), isLocatedIn(?d,?r), isProvidedBy(?e,?b), isWantedIn(?e,?r), serviceProvidedBy(?b,?d) → suggestDeviceActivation(?s,?d)*" et "[Rule 6] : *BasicService(?b), Device(?d), Effect(?e), Room(?r), System(?s), hasToProvide(?s,?e), isLocatedIn(?d,?r), isProvidedBy(?e,?b), isWantedIn(?e,?r), serviceProvidedBy(?b,?d) → suggestServiceActivation(?d,?b)*". En intégrant l'information de localisation issue du capteur IR01, le système peut lui proposer d'éteindre la lumière de la salle de bain de la même manière que dans l'étape 3. Cette proposition est acceptée par l'utilisateur : "[Log 12] : *Light5106.SwitchOFF*".

### 5.2.3 Résultats sur les jeux de données

Deux interprétations des résultats peuvent être réalisées en fonction des besoins de supervision. Si l'objectif est de réduire les coûts liés à l'installation et à la maintenance de capteurs, il est possible de limiter leur utilisation en remplaçant les données de certains d'entre eux par l'analyse sémantique des commandes envoyées par l'utilisateur. Si l'objectif est d'augmenter la sûreté de fonctionnement du système, il est possible d'utiliser notre approche de supervision en gardant les capteurs présents dans l'habitat pour avoir de la redondance d'information (i.e. un couple {*capteur, actionneur*}) lorsque l'utilisateur effectue une action sur son environnement domotique.

### 5.2.3.1 Limiter les coûts

Pour montrer l'intérêt d'analyser la sémantique des commandes envoyées par l'utilisateur, nous avons considéré les logs issus de capteurs comme des logs de commandes pour les équipements qui sont potentiellement composés du couple {*capteur*, *actionneur*}. L'illustration sur un extrait du jeu de données présentée dans la section précédente montre que notre approche de supervision permet de limiter l'utilisation de capteurs à 37.5%. Cela signifie que seulement trois capteurs sur huit sont utilisés pour assurer cette application de suivi de personnes (cf logs du tableau 5.4). Il s'agit du débitmètre *F101* et des détecteurs de mouvements infrarouges *IR01* et *IR02*) qui ne peuvent pas être directement associés<sup>12</sup> à un actionneur.

De manière générale, les actionneurs peuvent remplacer les capteurs électromagnétiques placés sur les portes ainsi que les contacteurs de lumières. Il est ainsi possible de limiter l'utilisation des capteurs de l'appartement Domus à 71% en moyenne pour la série 1 et 81% pour la série 2 (cf tableau 5.5). Ce résultat varie d'un utilisateur à un autre suivant son interaction avec les équipements domotiques. Par exemple, l'utilisation de capteurs pour l'utilisateur 5 (dans la série 1) peut être limitée à 59% : cela signifie qu'il est possible de limiter l'utilisation à 10 capteurs au lieu de 17 (les 7 autres peuvent être remplacés par des actionneurs dont les logs peuvent remplacer ceux des capteurs.). Les capteurs qui ne peuvent pas avoir d'actionneurs associés, comme les détecteurs de mouvements infrarouges ou les capteurs de pression pour la localisation de l'utilisateur, peuvent être malgré tout remplacés par des inférences logiques avec une moins bonne qualité de services (i.e. fréquence d'acquisition des données et fiabilité des connaissances inférées).

### 5.2.3.2 Augmenter la sûreté de fonctionnement

Dans un contexte privé de toute contrainte économique, la sûreté de fonctionnement peut être privilégiée. Si nous décidons de garder tous les capteurs présents dans l'appartement Domus, notre approche de supervision peut améliorer la sûreté de fonctionnement par redondance d'informations. Dans ces travaux, l'information "l'utilisateur interagit avec son système domotique" peut provenir de deux sources différentes : de données de capteurs (i.e. état de l'équipement) mais aussi de données d'actionneurs (i.e. commande envoyée). Cette redondance d'information est estimée à 67% pour l'illustration de la section précédente : 8/12 logs de capteurs (cf tableau 5.4) peuvent être associés à des requêtes utilisateur (cf logs avec \*).

---

12. Si les détecteurs de mouvements *IR01* et *IR02* donnent la localisation précise de l'utilisateur dans une partie ou l'autre de la cuisine, notre approche de supervision permet toutefois d'inférer la localisation de l'utilisateur à l'échelle d'une pièce lorsque ce dernier utilise une ressource d'activation locale.

|                      | Série 1    |            | Série 2    |            |
|----------------------|------------|------------|------------|------------|
|                      | Limitation | Redondance | Limitation | Redondance |
| <b>Utilisateur 1</b> | 79%        | 13%        | 82%        | 6%         |
| <b>Utilisateur 2</b> | 68%        | 10%        | 79%        | 9%         |
| <b>Utilisateur 3</b> | 75%        | 6%         | 90%        | 2%         |
| <b>Utilisateur 4</b> | 70%        | 9%         | 79%        | 6%         |
| <b>Utilisateur 5</b> | 59%        | 48%        | 81%        | 27%        |
| <b>Utilisateur 6</b> | 73%        | 12%        | 76%        | 6%         |
| <b>Moyenne</b>       | 71%        | 16%        | 81%        | 9%         |

Limitation : utilisation de capteurs / Redondance : information (actionneur ↔ capteur)

TABLE 5.5 – Résultats de l'application de supervision sur les jeux de données Domus

Sur l'ensemble des jeux de données Domus, 16% des informations issues de capteurs (série 1) ont pu être associées soit à une information de commande, soit au résultat d'une inférence logique pour déduire la localisation de l'utilisateur (cf tableau 5.5)<sup>13</sup>. La redondance d'informations dépend directement des interactions de l'utilisateur avec son environnement domotique : l'utilisateur 5 a par exemple beaucoup interagit avec les équipements domotiques (48% de redondances dans la première série). En revanche, lorsque l'utilisateur passe par exemple dans une pièce sans envoyer de commande domotique, aucune inférence logique n'est réalisée quant à sa localisation. L'approche de supervision étant pro-active, la fréquence d'acquisition des données dépend de l'utilisation du système domotique. Cette remarque permet d'expliquer la différence de résultats entre la première série et la deuxième. Dans la deuxième expérimentation (série 2), l'utilisateur répète la routine de la première série et réalise en plus une activité ne nécessitant pas de commande domotique. L'utilisation d'actionneurs passe alors de 29% à 19% entre les deux séries, ce qui provoque une diminution de la redondance d'informations (9% en moyenne pour la série 2).

13. Les résultats présentés dans le tableau 5.5 se basent sur les jeux de données bruts, sans nettoyage préalable des doublons involontaires de données de capteurs.



# 6

## Conclusion et perspectives

### 6.1 Résumé des contributions

#### 6.1.1 DSML et flot de conception domotique

La première contribution de cette thèse est un langage de modélisation spécifique au domaine de la domotique d'assistance (*Domain Specific Modeling Language*). Alors que les solutions existantes pour la configuration logicielle en domotique sont actuellement réservées à des experts / domoticiens du fait des procédures de bas-niveau utilisées et des concepts techniques à manipuler, nous avons cherché à intégrer une personne non-experte en domotique dans certaines des étapes de la conception. En effet, nous pensons que des membres de la famille ou des ergothérapeutes sont mieux placés que des domoticiens pour prendre en compte les besoins de vie exprimés par l'utilisateur en situation de handicap. Pour cela, nous définissons un *DSML* permettant de décrire à un haut niveau d'abstraction un système domotique sans avoir besoin de compétences techniques particulières. Ce dernier est composé de deux vues : une vue physique pour représenter l'environnement domotique par agrégation de composants et une vue fonctionnelle pour définir les interactions entre l'utilisateur et son système domotique. Dans cette dernière vue, deux types d'interactions sont définis. Ainsi, il est possible de définir des requêtes de différents types (i.e. services) comme des interdictions temporaires (i.e. modes) sur des ressources domotiques, offrant ainsi la possibilité de prendre en compte un grand nombre de besoins utilisateur.

La deuxième contribution de cette thèse est la proposition d'un flot de conception dirigé par les modèles. Basé sur l'IDM, il permet d'automatiser les différentes étapes de la conception domotique. Ce dernier est inspiré d'un processus de conception centré utilisateur pour intégrer un concepteur non-expert dans ces étapes de la conception. À partir de modèles indépendants de toute plateforme (*PIM*) décrivant l'environnement domotique et les interactions utilisateur-système définis à l'aide du *DSML*, un modèle spécifique à une plateforme est généré (*PSM*). Deux *PSM* ont ainsi été définis : un pour la partie commande relative à la plateforme domotique *Danah* et un pour la partie supervision basée sur une ontologie *OWL*.



Pour évaluer le *DSML* et le flot de conception, nous avons mené une série d'expérimentations contrôlées sur un cas d'étude réel : les appartements tremplins du centre de Kerpape. Dans la première expérimentation, nous avons appliqué le flot de conception sur ce cas d'étude pour aboutir à un système domotique à échelle réelle. Toutes les étapes du flot ont été déroulées sur l'appartement de 75m<sup>2</sup> équipé d'un bus domotique *KNX/EIB*. Nous avons réalisé ensuite une série d'expérimentations, auprès d'étudiants puis d'ergothérapeutes, pour intégrer des concepteurs non-experts en domotique en leur proposant de modéliser le système domotique du cas d'étude. Ces expérimentations et les sondages distribués aux participants à la fin de ces dernières ont permis de mettre en avant des résultats intéressants. Pour l'expérimentation contrôlée menée auprès d'étudiants, les taux de conformité des modèles produits sont de plus de 80% avec des temps jugés acceptables pour la définition de l'environnement domotique et pour la définition des interactions utilisateur-système du cas d'étude. Pour se rapprocher des conditions réelles d'utilisation, des expérimentations ont été réalisées avec deux ergothérapeutes du centre de Kerpape. Contrairement à l'expérimentation auprès d'étudiants, l'objectif était d'aboutir au fichier de configuration, pour un cas d'étude plus simple, avec un taux de conformité de 100%. Pour les deux personnes qui ont manipulé le logiciel, les objectifs ont été atteints avec des temps similaires, voire meilleurs que les temps passés par les étudiants lors de la première expérimentation.

Ces deux contributions ont donné lieu aux publications suivantes : [3], [4], [5] et [7] (cf publications personnelles). Nous considérons cette première partie suffisamment aboutie pour bénéficier d'un transfert technologique. L'entreprise Vity Technology, qui a porté un certain intérêt pour l'approche de conception proposée (spécifications logicielles définies par le client non-expert avant l'intégration domotique) reprendra ces travaux dans un but de commercialisation. Un travail de recherche sera ainsi mené dans le cadre du projet E-Santé *ASIM*, pour la définition d'une nouvelle syntaxe graphique. Cette perspective à court terme, présentée dans la section 6.2, peut se baser sur les résultats des expérimentations menées auprès d'étudiants et d'ergothérapeutes. Nous discutons par la suite de ces résultats.

### 6.1.2 Modèle de supervision non-intrusif

La troisième et dernière contribution de cette thèse est un modèle de supervision domotique non-intrusif basé sur les ontologies. Contrairement aux solutions existantes basées sur l'utilisation de capteurs, nous proposons d'analyser la sémantique des commandes domotiques envoyées par l'utilisateur. Pour cela, nous avons défini une ontologie modélisant les interactions entre l'utilisateur et son système domotique. L'ontologie et les règles qui lui sont associées permettent à un moteur d'inférences d'extraire des informations habituellement captées. Suivant la manière dont l'utilisateur interagit avec son système domotique, la localisation et l'intention de l'utilisateur peuvent être déduites à l'aide d'inférences logiques.

Pour valider notre approche, nous avons appliqué le modèle de supervision non-intrusif sur un jeu de données réel. Les résultats sur les jeux de données de l'appartement *Domus* de l'Université de Sherbrooke ont été interprétés de deux manières en fonction des besoins de supervision. Si l'objectif est de réduire les coûts liés à l'installation et à la maintenance de capteurs, il est possible de limiter leur utilisation en remplaçant les données de certains d'entre eux par l'analyse sémantique des commandes envoyées par l'utilisateur. L'utilisation de capteurs dans l'appartement *Domus* peut ainsi potentiellement être limitée à 71% en moyenne pour la série 1 et 81% pour la série 2, les capteurs électro-magnétiques placés sur les portes et les contacteurs de lumières pouvant être remplacés par des actionneurs. Si l'objectif est d'augmenter la sûreté de fonctionnement du système, il est possible d'utiliser notre approche de supervision en gardant les capteurs présents dans l'habitat pour avoir de la redondance d'information (i.e. un couple {capteur, actionneur}) lorsque l'utilisateur effectue une action sur son environnement domotique. La redondance d'information a été estimée à 16% en moyenne pour la série 1 du jeu de données *Domus* : cela signifie que 16% des informations issues de capteurs ont pu être associées soit à une information de commande, soit au résultat d'une inférence logique pour déduire la localisation de l'utilisateur.

Cette contribution a donné lieu aux publications suivantes : [1], [2] et [6] (cf publications personnelles). Nous considérons que les résultats obtenus pour cette partie sont suffisamment intéressants pour poursuivre les recherches sur ces travaux. Lorsque nous avons étudié l'intérêt des ontologies pour les systèmes domotiques d'assistance, quelques pistes de recherche ont pu être dégagées. Certaines de ces pistes, qui font majoritairement l'objet de perspectives à long terme, sont présentées dans la section 6.3.

## 6.2 Perspectives à court terme

### 6.2.1 Expérimentations contrôlées

Lorsqu'une expérimentation contrôlée est menée, il convient de discuter de la validité de la méthode mais aussi des résultats obtenus. Trois types de validité peuvent être étudiés et discutés dans notre cas :

- **Validité interne** : *“Quelle est la fiabilité et la certitude des conclusions obtenues ?”*  
Nous estimons que l'expérimentation réalisée ne comporte pas de biais importants en terme de validité interne. En effet, aucun des sujets n'a de connaissances en modélisation et/ou en conception de systèmes domotiques. Nous considérons à ce titre que les étudiants et les ergothérapeutes qui ont participé aux expérimentations ont un niveau de compétences identique dans ce domaine. Nous ne constatons pas non plus de biais évidents quant à l'instrumentation de cette expérience, puisque les documents fournis et l'outil de conception utilisé ont été les mêmes pour tous les participants.

- **Validité constructive** : “*Les mesures effectuées reflètent-elles ce que l’on souhaite étudier ?*” Si la variable  $T$  (temps) peut être sujette à des disparités du fait de l’appréhension subjective de l’outil de conception, l’étude de la variable  $C$  (conformité) peut en revanche être considérée comme un critère d’évaluation objectif. Dans le cas où l’on s’intéresse à la capacité des participants à traduire une description textuelle en modèle, le taux de conformité pris indépendamment du temps peut en effet être considéré comme objectif. Il en est évidemment de même lorsque l’objectif est d’avoir un taux de conformité de 100%.
- **Validité externe** : “*Les conclusions sont-elles généralisables ?*” Dans le cadre d’une expérimentation académique, il est souvent difficile de généraliser les résultats obtenus. Deux causes peuvent expliquer ce dernier point. La première concerne le faible nombre de participants qui pourrait être reproché à cette expérimentation contrôlée. La deuxième cause éventuelle porte sur le non-respect de l’hypothèse selon laquelle les sujets de l’étude ne sont pas assez représentatifs des personnes non-expertes ciblées en terme de connaissances/compétences en domotique. Pour diminuer l’impact de ces deux potentielles causes de biais expérimentaux, nous avons tenu à mener une expérimentation contrôlée avec des ergothérapeutes du centre de Kerpape (population cible). On peut toutefois remettre en cause le faible nombre de participants lors de cette expérimentation.

L’expérimentation contrôlée menée auprès d’ergothérapeutes a été pour nous l’occasion d’avoir des retours d’expérience de leur part. Au vu du faible nombre de participants, nous ne sommes pas en mesure de tirer des conclusions statistiques pertinentes pour cette population. Pour combler ce manque, des expérimentations plus significatives avec des ergothérapeutes du centre sont envisagées dans le cadre du projet ASIM. Ces expérimentations se dérouleront durant la phase d’évaluation du logiciel de configuration qui doit être développé et qui se base sur les travaux de cette thèse.

### 6.2.2 Syntaxe graphique pour le DSML

Les travaux autour du *DSML* présentés dans le chapitre 2 ont porté essentiellement sur la définition de la syntaxe abstraite (concepts à manipuler). Nous considérons la syntaxe concrète (i.e. graphique) comme une première proposition, qui nous a avant tout permis d’outiller le flot de conception pour évaluer une mise en situation réelle. Des résultats encourageants ont ainsi été relevés lors des expérimentations contrôlées (75% des étudiants qui ont participé à l’expérimentation ont jugé la traduction avec l’outil plus facile que la compréhension textuelle du tutoriel : une fois la description graphique comprise, la grande majorité trouve que l’outil est “*accessible et pas forcément difficile à manipuler*”). Nous sommes cependant conscients qu’un travail de recherche sur la définition d’une syntaxe graphique plus ergonomique doit être réalisé. Un transfert technologique du logiciel “prototype” développé dans ces travaux sera effectué dans le cadre du projet *ASIM*, à une fin de commercialisation. La syntaxe abstraite qui a fait l’objet de cette thèse sera conservée pour que les concepts introduits soient mis au service du plus grand nombre. En revanche, les expérimentations sur l’utilisabilité du

logiciel de conception et l'analyse préliminaire pour le projet *ASIM* ont montré l'intérêt de retravailler la syntaxe concrète (i.e. graphique). Entre autres, de nouvelles interactions sont à définir pour faciliter la définition des modèles d'environnement et d'interactions. La représentation graphique utilisée pour les scénarios doit par exemple être améliorée (l'intuitivité pour la définition de scénarios a été jugée par les étudiants à 2,25/4) pour rendre l'outil plus accessible à des concepteurs non-experts.

### 6.2.3 IHM dynamique et contextuelle

La modélisation des interactions utilisateur-système basée sur une ontologie permet à un moteur d'inférences d'interpréter les commandes envoyées par l'utilisateur pour lui proposer des services adaptés. La proposition dynamique et contextuelle de services est une contribution intéressante pour permettre d'adapter le fonctionnement de l'habitat à un utilisateur. Pour les personnes à mobilité réduite (e.g. paraplégique, tétraplégique) qui peuvent contrôler leur système domotique à partir d'un seul contact (e.g. bouton poussoir placé à un endroit physiquement accessible), un défilement automatique de services sur l'IHM est indispensable. Dans une telle situation, la sélection du service domotique désiré peut prendre beaucoup de temps. Nous pensons que l'approche de supervision proposée permettrait aux utilisateurs un accès plus rapide et plus facile aux services à partir de leur interface graphique (IHM). Pour cela, on peut envisager un menu d'accès rapide aux services qui sont proposés en fonction de la localisation et de l'intention de l'utilisateur. L'acceptation de services par l'utilisateur n'a pas pu être évaluée dans ces travaux de thèse compte tenu du fait que nous avons travaillé sur des jeux de données basés sur des capteurs. Aussi, un déploiement à échelle réelle dans les appartements tremplins permettrait d'évaluer l'acceptation des services proposés et de voir comment il est possible d'améliorer la vitesse d'accès aux services. Des métriques couramment utilisés dans le domaine de la prédiction de mots (taux d'économie de saisie ou *Keystroke Shift Reduction*) pourraient être relevées. Mais au delà de la proposition de services, il serait intéressant de voir comment les utilisateurs perçoivent et s'adaptent à ce genre de systèmes pro-actif.

## 6.3 Perspectives à long terme

### 6.3.1 Résolution de services de haut niveau sémantique

Les services de haut niveau sémantique, associés à une fonction, permettraient de définir des interactions intuitives pour l'utilisateur. Suivant les informations contextuelles (e.g. préférences, localisation), le système décide quelle ressource activer pour satisfaire les besoins de ce dernier. Par exemple le service *Allumer la lumière* (associé à la fonction du même nom) peut s'appliquer sur une lampe i) que l'utilisateur a l'habitude d'utiliser et ii) qui se trouve dans la même pièce que l'utilisateur. Cette prise de décision nécessite une résolution optimale pour activer une seule opération (e.g. allumer la lampe de chevet) à partir du service de haut niveau sémantique sollicité.

La résolution de ce type de services a été une piste sur laquelle nous avons également travaillé. Malheureusement, nous nous sommes vite rendu compte des limites d'un raisonnement basé uniquement sur des inférences logiques. Le système de règles permettant ces inférences devient très complexe lorsque l'on souhaite réaliser une résolution optimale (i.e. choix d'une seule ressource lors de l'activation de ce type de services). Aussi, nous pensons qu'une telle résolution doit s'inscrire dans une aide à la décision multi-critères. La résolution de ce type de services peut par exemple garantir à la fois une qualité de services (perçue par l'utilisateur) acceptable tout en minimisant la consommation énergétique globale de l'habitat. Pour prendre en compte ces critères, la multiplicité des sources d'informations et leur caractère imprécis, nous pensons qu'il serait pertinent d'intégrer de la logique floue ou probabiliste, en plus du processus de raisonnement logique proposé dans ces travaux. Il est d'ailleurs possible de mélanger ces deux méthodes d'inférences (floues et probabilistes), tel que le propose la Théorie de Dempster-Shafer (DST). Le principal intérêt de ce type de modélisation est qu'il permet de rechercher un consensus entre plusieurs sources d'information distinctes ou plusieurs objectifs antagonistes tels qu'ils peuvent apparaître dans la gestion dynamique d'un habitat domotisé. Cette direction est celle retenue pour le projet DyDom, un travail de thèse qui s'inscrit dans la continuité de ces travaux.

### 6.3.2 QoS pour le déploiement de capteurs

L'application pour le suivi de personnes et les résultats sur les jeux de données Domus complets ont pu montrer l'intérêt d'analyser la sémantique des commandes pour inférer des informations pertinentes sur l'utilisateur. En fonction du contexte d'utilisation, il est possible i) de limiter l'utilisation de capteurs ou ii) d'augmenter la redondance d'information. Pour cela, notre approche considère les logs de commande ainsi que les inférences logiques i) à la place ou ii) en ajout de contacteurs et de détecteurs de mouvement. Dans le cadre de ces travaux de thèse, nous avons volontairement choisi de montrer deux utilisations possibles qui sont opposées, mais il doit en réalité exister un grand nombre de solutions intermédiaires.

Pour aider à choisir le nombre de capteurs qu'il serait possible de déployer, un modèle de *QoS* (*Quality of Service*) doit être défini. Intuitivement, nous pensons que la *QoS* pour la supervision de systèmes domotiques est liée à la fréquence d'acquisition des données (capteurs et/ou actionneurs) mais aussi à la fiabilité des informations inférées. L'utilisation de notre approche sans capteurs permet d'aboutir à une supervision avec une *QoS* (*Quality of Service*) faible. En effet, la fréquence d'acquisition peut être très faible si l'on ne fait pas une utilisation pro-active du système domotique. À l'inverse, l'utilisation de l'approche en parallèle des capteurs permet d'augmenter la *QoS* par redondance d'informations. Par exemple, lorsque l'utilisateur ouvre une porte, un log de commande et un log de capteur peuvent être fusionnés pour fiabiliser l'information d'ouverture de porte. Plusieurs niveaux de *QoS* peuvent ainsi être définis :

- niveau 1 : pas de capteurs, mais uniquement l'approche de supervision

- niveau 2 : quelques capteurs en plus de l'approche de supervision
- niveau 3 : tous les capteurs en plus de l'approche de supervision

Il doit exister un large éventail de solutions de déploiement de capteurs avec des QoS différentes pour la supervision d'un système domotique. Une modélisation plus précise de cette QoS permettrait de dimensionner dès la phase de conception le nombre de capteurs à déployer avec l'approche de supervision proposée dans ces travaux. Il serait ainsi possible de préciser les niveaux intermédiaires qui dérivent du niveau 2 en répondant à la question : *"Combien de capteurs doivent être utilisés pour assurer une QoS pour la supervision domotique idéale pour l'utilisateur ?"*. Un outil original pourrait être conjointement développé pour simuler les déplacements de l'utilisateur, les détections de capteurs mais aussi les activations de services. Cet outil pourrait dans un premier temps permettre de définir le modèle de QoS de manière empirique. Dans un deuxième temps, il pourrait servir pour valider le nombre minimum de capteurs à déployer pour assurer la supervision d'un habitat domotisé.

### 6.3.3 Aide technique mobile unifiée

Au delà de la domotique, nous pensons que d'autres types d'aides techniques électroniques et informatiques peuvent bénéficier de l'approche de supervision non-intrusive basée sur les ontologies proposée dans ces travaux. Nous portons un intérêt particulier aux logiciels d'aide à la communication. Ce type d'aides techniques permet à des personnes dépourvues de la parole, soit temporairement, soit définitivement, de communiquer à l'aide d'une synthèse vocale. Pour cela, un clavier virtuel alphabétique et/ou pictographique est affiché à l'écran. L'intervention de l'utilisateur se limite donc à la désignation des caractères ou pictogrammes désirés pour composer sa phrase. Pour des personnes n'ayant pas les capacités gestuelles de contrôler le curseur de la souris, un système de défilement automatique peut être mis en place pour sélectionner ces symboles à partir d'un bouton poussoir. La prédiction de mots permet finalement d'accélérer la vitesse d'écriture. L'utilisation de ce type de logiciel se fait bien souvent en parallèle du logiciel de domotique sur deux interfaces différentes. Nous pensons logiquement que ces deux aides techniques, couramment utilisées par des patients du centre de Kerpape, doivent être unifiées.

Pour aller plus loin, nous sommes persuadés que l'unification de ces deux aides techniques permettrait d'améliorer à la fois la vitesse d'écriture (pour la communication) et la proposition de services (pour la domotique). L'analyse sémantique des commandes domotiques telle que proposée dans ces travaux permettrait de contextualiser le logiciel d'aide à la communication. Les pictogrammes proposés pourraient être contextualisés en fonction des commandes domotiques envoyées : ils pourraient être différents suivant la localisation inférée de l'utilisateur. Dans un centre de rééducation, on y retrouverait par exemple en premier plan le nom des thérapeutes et des phrases courantes qui n'ont que peu d'utilité dans un autre contexte : *"ma chambre est la 108"*,

*“j’ai une séance d’orthophonie”*, etc. Les informations de localisation inférées à partir des commandes domotiques pourraient aussi servir à affiner la prédiction de mots. Dans le sens inverse, l’utilisation du logiciel d’aide à la communication peut aider à recueillir des informations sur l’utilisateur et permettre l’adaptation de services domotiques en conséquence. Par exemple, lorsque l’utilisateur sélectionne le pictogramme *“j’ai envie d’aller aux toilettes”*, le système domotique pourrait ouvrir la porte et allumer la lumière de la salle de bain.

## 6.4 Le mot de la fin

Parmi les perspectives mentionnées précédemment, j’aimerais citer dans cette section celles qui me tiennent particulièrement à coeur et qui me semblent prioritaires par rapport à mon nouveau contexte professionnel. Au laboratoire d’électronique du centre de Kerpape, nous avons pour mission d’adapter ou de développer (lorsque cela est nécessaire) des aides techniques électroniques et informatiques pour les patients du centre. Cela comprend un large spectre qui balaie l’aide à la communication, l’accès à l’ordinateur, le contrôle d’environnement (i.e. la domotique) ou encore l’aide au déplacement. Toutes ces aides ne sont actuellement pas présentes dans une seule et unique interface, mais se retrouvent au contraire sur plusieurs interfaces dédiées. Il s’agit pourtant de fonctions essentielles pour réaliser les activités de la vie quotidienne. Je suis convaincu que la proposition d’une interface unifiée regroupant la domotique et l’aide à la communication permettrait de simplifier et d’améliorer l’utilisation de ces deux aides techniques. L’électronique et l’informatique y sont utilisées comme des moyens de compenser une situation de handicap en permettant le contrôle à distance des équipements ou la communication assistée par ordinateur. Ce contexte numérique riche, dans lequel évoluent les patients du centre, permet également l’automatisation : la définition de scénarios pour la domotique ou la prédiction de mots pour l’aide à la communication. Mais ce contexte numérique est une source d’informations qui à mon sens est encore sous-exploitée. En unifiant ces deux aides techniques, je suis persuadé qu’un grand nombre d’informations pertinentes sur l’utilisateur peut être recueilli pour permettre une adaptation dynamique et contextuelle. Aussi, je pense que le déploiement à échelle réelle de notre modèle de supervision est une perspective à plus court terme particulièrement intéressante pour faire ce premier pas dans le domaine de la domotique.

# Bibliographie

- [Abowd *et al.* 2002] G.D. Abowd, A.F. Bobick, I.A. Essa, E.D. Mynatt et W.A. Rogers. *The aware home : A living laboratory for technologies for successful aging*. In Proceedings of the AAAI-02 Workshop - Automation as Caregiver, pages 1–7, 2002.
- [Adami *et al.* 2003] AM Adami, TL Hayes et M. Pavel. *Unobtrusive monitoring of sleep patterns*. In Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE, volume 2, pages 1360–1363. Ieee, 2003.
- [Ameisen *et al.* 2011] B. Héritier F. Hunyadi M. Ameisen J.C. Heilbrunnet *al.* *L'éternel singulier. questions autour du handicap*. Le Bord de l'eau, 2011.
- [Baader 2003] F. Baader. *The description logic handbook : theory, implementation, and applications*. Cambridge Univ Pr, 2003.
- [Basili & Weiss 1984] V.R. Basili et D.M. Weiss. *A methodology for collecting valid software engineering data*. Software Engineering, IEEE Transactions on, no. 6, pages 728–738, 1984.
- [Bauer & Steiner 2009] R. Bauer et M. Steiner. *Injuries in the European Union : Statistics Summary 2005-2007*. Vienna, Eurosafe, KfV, 2009.
- [Belabbas *et al.* 2006a] A. Belabbas, P. Berruet, A. Rossi et J. Luc Philippe. *A modeling approach to control a handicap technical assistance system*. WSEAS Transactions on Information Science and Applications, vol. 3, no. 8, pages 1460–1467, 2006.
- [Belabbas *et al.* 2006b] A. Belabbas, P. Berruet, A. Rossi, J.L. Philippeet *al.* *Cooperative model generation for disabled people assistance*. In Proceedings of the 5th WSEAS international conference on Telecommunications and informatics, pages 438–445. World Scientific and Engineering Academy and Society (WSEAS), 2006.
- [Belabbas 2007] A. Belabbas. *Méthodes de conception de systèmes contrôle/commande reconfigurables en présence d'aléas : application au handicap*. PhD thesis, Laboratoire d'Électronique des systèmes Temps Réel, Université de Bretagne-Sud, 2007.
- [Betremieux *et al.* 2010] N. Betremieux, J. Fabre et T. Vacher. *Le Handicap à travers ses multiples approches*, INSEE, 2010.
- [Bézivin 2004] J. Bézivin. *In search of a basic principle for model driven engineering*. Novatica Journal, Special Issue, pages 21–24, 2004.
- [Bézivin 2005] J. Bézivin. *On the unification power of models*. Software and Systems Modeling, vol. 4, no. 2, pages 171–188, 2005.
- [Bhattacharya & Das 2002] A. Bhattacharya et S.K. Das. *LeZi-update : An information-theoretic framework for personal mobility tracking in PCS networks*. Wireless Networks, vol. 8, no. 2/3, pages 121–135, 2002.
- [Bittencourt *et al.* 2009] I.I. Bittencourt, P. Bispo, E. Costa, J. Pedro, D. Veras, D. Derrmeval et H. Pacca. *Modeling jade agents from gaia methodology under the perspective of semantic web*. Enterprise Information Systems, pages 780–789, 2009.



- [Bodenheimer *et al.* 2002] T. Bodenheimer, K. Lorig, H. Holman et K. Grumbach. *Patient self-management of chronic disease in primary care*. JAMA : the journal of the American Medical Association, vol. 288, no. 19, pages 2469–2475, 2002.
- [Bonhomme 2008] Sylvain Bonhomme. *Méthodologie et outils pour la conception d'un habitat intelligent*. PhD thesis, Institut National Polytechnique de Toulouse, 2008.
- [Bonino & Corno 2008] D. Bonino et F. Corno. *Dogont-ontology modeling for intelligent domotic environments*. The Semantic Web-ISWC 2008, pages 790–803, 2008.
- [Bonino *et al.* 2009] D. Bonino, E. Castellina et F. Corno. *Automatic domotic device interoperation*. Consumer Electronics, IEEE Transactions on, vol. 55, no. 2, pages 499–506, 2009.
- [Boulos *et al.* 2007] M. Boulos, A. Rocha, A. Martins, M. Vicente, A. Bolz, R. Feld, I. Tchoudovski, M. Braeckleinet *al.* *CAALYX : a new generation of location-based services in healthcare*. International Journal of Health Geographics, vol. 6, no. 1, page 9, 2007.
- [Brickley & Miller 2000] D. Brickley et L. Miller. *FOAF vocabulary specification 0.91*. Rapport technique, Tech. rep. ILRT Bristol, Nov. 2007. ur l : <http://xmlns.com/foaf/spec/20071002.html>, 2000.
- [Bricon-Souf & Newman 2007] N. Bricon-Souf et C.R. Newman. *Context awareness in health care : A review*. international journal of medical informatics, vol. 76, no. 1, pages 2–12, 2007.
- [Brody 1985] H. Brody. *Smartest house on the block*. High Technol. (United States), vol. 5, no. 5, 1985.
- [Brookmeyer *et al.* 2007] R. Brookmeyer, E. Johnson, K. Ziegler-Graham et H.M. Arighi. *Forecasting the global burden of Alzheimer's disease*. Alzheimers Dement, vol. 3, no. 3, pages 186–191, 2007.
- [Chan *et al.* 2008] M. Chan, D. Estève, C. Escriba et E. Campo. *A review of smart homes - Present state and future challenges*. Computer Methods and Programs in Biomedicine, vol. 91, no. 1, pages 55–81, 2008.
- [Chan *et al.* 2009] M. Chan, E. Campo, D. Estève et J.Y. Fourniols. *Smart homes - Current features and future perspectives*. Maturitas, vol. 64, no. 2, pages 90–97, 2009.
- [Chen *et al.* 2005] H. Chen, T. Finin et A. Joshi. *The SOUPA ontology for pervasive computing*. Ontologies for agents : Theory and experiences, pages 233–258, 2005.
- [Chikhaoui *et al.* 2010] B. Chikhaoui, S. Wang et H. Pigot. *A New Algorithm Based On Sequential Pattern Mining For Person Identification In Ubiquitous Environments*. In Proc. of the Fourth Int. Workshop on Knowledge Discovery from Sensor Data (ACM SensorKDD'10), Washington, DC, July, pages 25–28, 2010.
- [Chossy 2006] J.F. Chossy. *Rapport d'information n 3161 déposé par la commission des Affaires culturelles, familiales sur la mise en application de la loi n 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées*, 2006.

- [Colomb *et al.* 2006] R. Colomb, K. Raymond, L. Hart, P. Emery, C. Welty, G. Xie et E. Kendall. *The object management group ontology definition metamodel*. *Ontologies for Software Engineering and Software Technology*, pages 217–247, 2006.
- [Consulting & Deere 2005] D.F. Consulting et J. Deere. *Ontology Definition Metamodel*. 2005.
- [Cook *et al.* 2003] D.J. Cook, M. Youngblood, E.O. Heierman III, K. Gopalratnam, S. Rao, A. Litvin et F. Khawaja. *MavHome : An agent-based smart home*. In *Pervasive Computing and Communications, 2003.(PerCom 2003)*. Proceedings of the First IEEE International Conference on, pages 521–524. IEEE, 2003.
- [Czarnecki & Helsen 2003] K. Czarnecki et S. Helsen. *Classification of model transformation approaches*. In *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, volume 45, pages 1–17. Citeseer, 2003.
- [Czarnecki & Helsen 2006] K. Czarnecki et S. Helsen. *Feature-based survey of model transformation approaches*. *IBM Systems Journal*, vol. 45, no. 3, pages 621–645, 2006.
- [Das *et al.* 2002] S.K. Das, D.J. Cook, A. Battacharya, E.O. Heierman III et T.Y. Lin. *The role of prediction algorithms in the MavHome smart home architecture*. *Wireless Communications, IEEE*, vol. 9, no. 6, pages 77–84, 2002.
- [Davis 2003] J. Davis. *GME : the generic modeling environment*. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 82–83. ACM, 2003.
- [De Lamotte *et al.* 2007] F.F. De Lamotte, P. Berruet et J.L. Philippe. *Using model engineering for the criticality analysis of reconfigurable manufacturing systems architectures*. *International journal of manufacturing technology and management*, vol. 11, no. 3, pages 315–337, 2007.
- [De Lamotte *et al.* 2008] F. De Lamotte, JP Departe, F. Le Saout, J.P. Diguët et JL Philippe. *Quatra : Final report (Technical report, in french)*. *Lab-STICC*. See demo here : <http://www.youtube.com/watch?v=T6GCFnkLTc0>, Juin 2008.
- [Demongeot 2005] J. Demongeot. *Technologie et soins g erontologiques : les implications  ethiques et les r eflexions pour le futur*. *G erontologie et soci et e*, no. 2, pages 121–136, 2005.
- [Diener & Wong 2009] H. Diener et P. Wong. *Developments in Secondary Stroke Prevention*. *European Neurological Review - Volume 3 Issue 2*, 2009.
- [Domus ] Domus. *Monitoring Dataset*. <http://domus.usherbrooke.ca/jeux-de-donnees/> (derni ere visite le 26 juin 2012).
- [Dupr e & Karjalainen 2003] D. Dupr e et A. Karjalainen. *Employment of disabled people in Europe in 2002*. *Statistics in Focus : Population and Social Conditions*. Theme, pages 3–26, 2003.
- [Echelon Corporation 1998] Echelon Corporation. *LonMaker Integration Tool*. Official website : <http://www.echelon.com/>, 1998.

- [Eclipse 2007] IDE Eclipse. *The Eclipse Foundation*, 2007.
- [Elite Care ] Elite Care. *Assisted Living and Memory Care in Portland*. <http://www.elite-care.com> (dernière visite le 23 juin 2012).
- [Espagnacq 2012] M. Espagnacq. *Dossier Solidarité et Santé - Personnes ayant des limitations fonctionnelles physiques - Panorama des aides et aménagements du logement*, 2012.
- [Falleri et al. 2006] J.R. Falleri, M. Huchard, C. Nebutet al. *Towards a traceability framework for model transformations in kermeta*. 2006.
- [Favre et al. 2006] J.M. Favre, J. Estublier et M. Blay-Fornarino. *L'ingénierie dirigée par les modèles au-delà du mda*. Hermès Science Publications Paris, 2006.
- [Fleurey et al. 2006] F. Fleurey, Z. Drey, D. Vojtisek, C. Faucher et V. Mahé. *Kermeta Language, Reference Manual*. Internet : <http://www.kermeta.org/docs/KerMeta-Manual.pdf>. IRISA, 2006.
- [Fougeyrollas & RIPPH 1998] P. Fougeyrollas et RIPPH. *Classification québécoise-processus de production du handicap*. RIPPH= SCCIDIH, 1998.
- [Fournier-Viger 2005] P Fournier-Viger. *Une introduction aux logiques de description*, 2005.
- [Friedenthal et al. 2011] S. Friedenthal, A. Moore et R. Steiner. *A practical guide to sysml : the systems modeling language*. Morgan Kaufmann, 2011.
- [GerHome ] GerHome. *GerHome website*.
- [Gouin-Vallerand et al. 2012] C. Gouin-Vallerand, P. Roy, B. Abdulrazak, S. Giroux et A. Dey. *A Macro and Micro Context Awareness Model for the Provision of Services in Smart Spaces*. *Impact Analysis of Solutions for Chronic Disease Prevention and Management*, pages 173–181, 2012.
- [Greenfield & Short 2003] J. Greenfield et K. Short. *Software factories : assembling applications with patterns, models, frameworks and tools*. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 16–27. ACM, 2003.
- [Gronback 2009] R.C. Gronback. *Eclipse modeling project : a domain-specific language toolkit*. Addison-Wesley Professional, 2009.
- [Gruber et al. 1995] T.R. Gruber et al. *Toward principles for the design of ontologies used for knowledge sharing*. *International Journal of Human Computer Studies*, vol. 43, no. 5, pages 907–928, 1995.
- [Habitat Sénior 2012] Habitat Sénior. *Habitat Sénior*. [http://www.habitat-senior.com/Info\\_Habitat\\_Ergonomie\\_Intro.html](http://www.habitat-senior.com/Info_Habitat_Ergonomie_Intro.html) (dernière visite le 14 juillet 2012), 2012.
- [Hamani et al. 2004] N. Hamani, N. Dangoumau et E. Craye. *A formal approach for reactive mode handling*. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, pages 4306–4311, 2004.

- [Hamani *et al.* 2006] N. Hamani, N. Danguoumau et E. Craye. *A functional modeling approach for mode handling of flexible manufacturing systems*. In Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing, pages 271–276, 2006.
- [Haux 2006] R. Haux. *Individualization, globalization and health-about sustainable information technologies and the aim of medical informatics*. International journal of medical informatics, vol. 75, no. 12, pages 795–808, 2006.
- [Helal *et al.* 2005] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura et E. Jansen. *The gator tech smart house : A programmable pervasive space*. Computer, vol. 38, no. 3, pages 50–60, 2005.
- [Hillairet *et al.* 2008] G. Hillairet, F. Bertrand et J.Y. Lafaye. *MDE for publishing Data on the Semantic Web*. In Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS. Citeseer, 2008.
- [Hobbs *et al.* 2002] J.R. Hobbs, G. Ferguson, J. Allen, P. Hayes, I. Niles et A. Pease. *A DAML ontology of time*. online : <http://www.cs.rochester.edu/~ferguson/daml/daml-time-20020830.txt>, 2002.
- [Horridge & Bechhofer 2011] M. Horridge et S. Bechhofer. *The owl api : A java api for owl ontologies*. Semantic Web, vol. 2, no. 1, pages 11–21, 2011.
- [Horrocks *et al.* 2004] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, M. Dean *et al.* *SWRL : A semantic web rule language combining OWL and RuleML*. W3C Member submission, vol. 21, page 79, 2004.
- [INSEE 2010a] INSEE. *Le handicap à travers ses multiples approches*, 2010.
- [INSEE 2010b] INSEE. *Les situations de handicap, une question d'âge*, 2010.
- [INSEE 2012] INSEE. *Dépenses courantes de santé en 2011*. website : <http://www.insee.fr/> (dernière visite le 22 octobre 2012), 2012.
- [Intille *et al.* 2002] S. Intille, C. Kukla et X. Ma. *Eliciting user preferences using image-based experience sampling and reflection*. In CHI'02 extended abstracts on Human factors in computing systems, pages 738–739. ACM, 2002.
- [Intille 2002] S.S. Intille. *Designing a home of the future*. Pervasive Computing, IEEE, vol. 1, no. 2, pages 76–82, 2002.
- [ISEE 2010] ISEE. *IGEPv2 Board*. Official website : <http://www.igep.es/>, 2010.
- [Jeuland 2009] F.X. Jeuland. *La maison communicante. réussir son installation domotique et multimédia* éd. 2010 (3 éd.). 2009.
- [Jimenez *et al.* 2009] M. Jimenez, F. Rosique, P. Sanchez, B. Alvarez et A. Iborra. *Habitation : a domain-specific language for home automation*. IEEE Software, vol. 26, no. 4, pages 30–38, 2009.
- [Jouault & Bézivin 2006] F. Jouault et J. Bézivin. *KM3 : a DSL for Metamodel Specification*. Formal Methods for Open Object-Based Distributed Systems, pages 171–185, 2006.

- [Jouault & Kurtev 2006] F. Jouault et I. Kurtev. *Transforming models with ATL*. In Satellite Events at the MoDELS 2005 Conference, pages 128–138. Springer, 2006.
- [Jouault *et al.* 2006] F. Jouault, F. Allilaire, J. Bézivin, I. Kurtev et P. Valduriez. *ATL : a QVT-like transformation language*. In Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, page 720. ACM, 2006.
- [Kadouche *et al.* 2010] R. Kadouche, H. Pigot, B. Abdulrazaka et S. Giroux. *Support vector machines for inhabitant identification in smart houses*. Ubiquitous Intelligence and Computing, pages 83–95, 2010.
- [Kastner *et al.* 2005] W. Kastner, G. Neugschwandtner et M. K"ogler. *An open approach to EIB/KNX software development*. Citeseer, 2005.
- [Knublauch *et al.* 2004] H. Knublauch, R. Ferguson, N. Noy et M. Musen. *The Protégé OWL plugin : An open development environment for semantic web applications*. The Semantic Web–ISWC 2004, pages 229–243, 2004.
- [KNX Association 1993] KNX Association. *Engineering Tool Software*. Official website : <http://www.knx.org/>, 1993.
- [KNX 2003] KNX. *KNX, Le standard mondial pour la gestion technique des bâtiments résidentiels & tertiaires*, 2003.
- [Koppel 2002] R. Koppel. *Alzheimer's disease : The costs to US businesses in 2002*. Washington, DC : Alzheimer's Association, 2002.
- [Lankri *et al.* 2008] S. Lankri, P. Berruet, A. Rossi et J.L. Philippe. *Architecture and models of the DANAH assistive system*. In Proceedings of the 3rd international workshop on Services integration in pervasive environments, pages 19–24. ACM, 2008.
- [Lankri *et al.* 2009] S. Lankri, P. Berruet et J.L. Philippe. *Service Reconfiguration in the DANAH Assistive System*. Ambient Assistive Health and Wellness Management in the Heart of the City, pages 269–273, 2009.
- [Lankri 2009] S. Lankri. *Services et Navigation pour Personnes Dépendantes en Environnement domotique*. PhD thesis, Laboratoire en Sciences et Techniques de l'Information, de la Communication et de la Connaissance, Université de Bretagne-Sud, 2009.
- [Le Gloahec 2011] V. Le Gloahec. *Un langage et une plateforme pour la définition et l'exécution de bonnes pratiques de modélisation*. PhD thesis, Laboratoire de Recherche Informatique et des Applications de Vannes et Lorient, Université de Bretagne-Sud, 2011.
- [Lézoray *et al.* 2011] J.B. Lézoray, M.T. Segarra, A. Phung-Khac, A. Thépaut, J.M. Gilliot et A. Beugnard. *A design process enabling adaptation in pervasive heterogeneous contexts*. Personal and Ubiquitous Computing, vol. 15, no. 4, pages 353–363, 2011.
- [Liao & Tu 2007] H.C. Liao et C.C. Tu. *A RDF and owl-based temporal context reasoning model for smart home*. Information Technology Journal, vol. 6, no. 8, pages 1130–1138, 2007.

- [Maguire 2001] M. Maguire. *Methods to support human-centred design*. International journal of human-computer studies, vol. 55, no. 4, pages 587–634, 2001.
- [McDonald et al. ] HA McDonald, CD Nugent, G. Moore, DD. Finlay et J. Hallberg. *A web based tool for storing and visualising data generated within a smart home*.
- [McGuinness et al. 2004] D.L. McGuinness, F. Van Harmelen et al. *OWL web ontology language overview*. W3C recommendation, vol. 10, pages 2004–03, 2004.
- [Meixner et al. 2011] G. Meixner, M. Seissler et K. Breiner. *Model-Driven Useware Engineering*. Model-Driven Development of Advanced User Interfaces, pages 1–26, 2011.
- [Mernik et al. 2005] M. Mernik, J. Heering et A.M. Sloane. *When and how to develop domain-specific languages*. ACM computing surveys (CSUR), vol. 37, no. 4, pages 316–344, 2005.
- [Mihailidis et al. 2004] A. Mihailidis, B. Carmichael et J. Boger. *The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home*. Information Technology in Biomedicine, IEEE Transactions on, vol. 8, no. 3, pages 238–247, 2004.
- [Miller et al. 2001] J. Miller, J. Mukerjiet al. *Model driven architecture (mda)*. Object Management Group, Draft Specification ormsc/2001-07-01, 2001.
- [MOF 2002] MOF. *Meta Object Facility Specification*. 2002.
- [Moore 1999] M. Moore. *The evolution of telemedicine*. Future Generations in Computer Systems, vol. 15, no. 2, pages 245–254, 1999.
- [Mozer 1998] M.C. Mozer. *The Neural Network House : An Environment hat Adapts to its Inhabitants*. In Proc. AAAI Spring Symp. Intelligent Environments, 1998.
- [Muller et al. 2005] P.A. Muller, F. Fleurey et J.M. Jézéquel. *Weaving executability into object-oriented meta-languages*. Model Driven Engineering Languages and Systems, pages 264–278, 2005.
- [Muñoz & Pelechano 2005] J. Muñoz et V. Pelechano. *Building a software factory for pervasive systems development*. In Advanced Information Systems Engineering, pages 342–356. Springer, 2005.
- [Muñoz et al. 2006] J. Muñoz, E. Serral, C. Cetina et V. Pelechano. *Applying a Model-Driven Method to the Development of a Pervasive Meeting Room*. ERCIM News, vol. 65, pages 44–45, 2006.
- [Musset et al. 2006] J. Musset, E. Juliot, S. Lacrampe, W. Piers, C. Brun, L. Goubet, Y. Lussaud et F. Allilaire. *ACCELEO User Guide*. 2006.
- [Nagai et al. 2009] M. Nagai, S. Hoshide et K. Kario. *Hypertension and dementia*. American journal of hypertension, vol. 23, no. 2, pages 116–124, 2009.
- [Nain et al. 2008] G. Nain, E. Daubert, O. Barais et J.M. Jézéquel. *Using mde to build a schizophrenic middleware for home/building automation*. Towards a Service-Based Internet, pages 49–61, 2008.

- [Nisbet 1996] P. Nisbet. *Integrating assistive technologies : current practices and future possibilities*. Medical engineering & physics, vol. 18, no. 3, pages 193–202, 1996.
- [Nourizadeh *et al.* 2009] S. Nourizadeh, C. Deroussent, Y.Q. Song, J.P. Thomesse *et al.* *Medical and home automation sensor networks for senior citizens telehomecare*. 2009.
- [Noury *et al.* 2003] N. Noury, G. Virone, J. Ye, V. Rialle *et al.* *New trends in health smart homes*. ITBM-RBM (RBM), vol. 24, no. 3, pages 122–135, 2003.
- [Nugent *et al.* ] C. Nugent, D. Finlay, R. Davies, H. Wang, H. Zheng, J. Hallberg, K. Synnes *et al.* *homeML—an open standard for the exchange of data within smart environments*.
- [Obeo 2012] Obeo. *Obeo Designer 6*, 2012.
- [OMG 2006] OMG. *MOF 2.0 core specification*. <http://www.omg.org/spec/MOF/2.0>, 2006.
- [Paas *et al.* 2004] F. Paas, A. Renkl *et al.* *Cognitive load theory : Instructional implications of the interaction between information structures and cognitive architecture*. Instructional Science, vol. 32, no. 1, pages 1–8, 2004.
- [Pardillo & Cachero 2010] J. Pardillo *et al.* *Domain-specific language modelling with UML profiles by decoupling abstract and concrete syntaxes*. Journal of Systems and Software, vol. 83, no. 12, pages 2591–2606, 2010.
- [Paré *et al.* 2007] G. Paré, M. Jaana *et al.* *Systematic review of home telemonitoring for chronic diseases : the evidence base*. Journal of the American Medical Informatics Association, vol. 14, no. 3, pages 269–277, 2007.
- [Petri 1962] C. A. Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für instrumentelle Mathematik, Bonn, 1962.
- [PIPAME 2009] PIPAME. *Pôle interministériel de prospectives et d'anticipation des mutations économiques - Diffusion des nouvelles technologies (NTE) dans le bâtiment*. 2009.
- [Pruski 2003] A. Pruski. *Assistance technique au handicap (Traité IC2, série Systèmes automatisés)*. Recherche, vol. 67, page 02, 2003.
- [Reinisch *et al.* 2009] C. Reinisch, W. Granzer, F. Praus *et al.* *Integration of heterogeneous building automation systems using ontologies*. In Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE, pages 2736–2741. IEEE, 2009.
- [Robert-Bobée 2006] I. Robert-Bobée. *principal : Projections de population pour la France métropolitaine à l'horizon 2050 : la population continue de croître et le vieillissement se poursuit*. 2006.
- [Román *et al.* 2002] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R.H. Campbell *et al.* *A middleware infrastructure for active spaces*. Pervasive Computing, IEEE, vol. 1, no. 4, pages 74–83, 2002.
- [Sanchez *et al.* 2011] P. Sanchez, M. Jimenez, F. Rosique, B. Alvarez *et al.* *A framework for developing home automation systems : from requirements to code*. Journal of Systems and Software, 2011.

- [Schlender 2003] B. Schlender. *Intel's Andy Grove : The next battles in tech*. Fortune, May, page 16, 2003.
- [Schmidt-Schaufl & Smolka 1991] M. Schmidt-Schaufl et G. Smolka. *Attributive concept descriptions with complements*. Artificial intelligence, vol. 48, no. 1, pages 1–26, 1991.
- [Shearer et al. 2008] R. Shearer, B. Motik et I. Horrocks. *HermiT : A highly-efficient OWL reasoner*. In Proceedings of the 5th International Workshop on OWL : Experiences and Directions (OWLED 2008), pages 26–27, 2008.
- [SIGAAL ] SIGAAL. *SIGAAL website*.
- [Steinberg et al. 2008] D. Steinberg, F. Budinsky, E. Merks et M. Paternostro. *Eclipse modeling framework*. Addison-Wesley Professional, 2008.
- [SysML 2008] OMG SysML. *1.0 Specification*. Object Management Group, 2008.
- [Szyperski et al. 2002] C. Szyperski, D. Gruntz et S. Murer. *Component software : beyond object-oriented programming*. 2002, 2002.
- [Telecom 2007] F. Telecom. *SmartQVT : An open source model transformation tool implementing the MOF 2.0 QVT-Operational language*, 2007, 2007.
- [Toguyéni et al. 2003] A.K.A. Toguyéni, P. Berruet et E. Craye. *Models and algorithms for failure diagnosis and recovery in FMSs*. International Journal of Flexible Manufacturing Systems, vol. 15, no. 1, pages 57–85, 2003.
- [Truong et al. 2009a] T.B.T. Truong, F.F. de Lamotte, J. Digué et al. *Proactive remote healthcare based on multimedia and home automation services*. In Proceedings of the 5th IEEE CASE, pages 385–390, 2009.
- [Truong et al. 2009b] T.B.T. Truong, F. Frizon de Lamotte, J.P. Digué et F. Saïd-Hocine. *Assisted living service identification based on activity patterns*. In Pervasive Computing (JCPC), 2009 Joint Conferences on, pages 417–422. IEEE, 2009.
- [Truong et al. 2010] T.B.T. Truong, F. Frizon de Lamotte, J.P. Digué et F. Saïd-Hocine. *Alert management for home healthcare based on home automation analysis*. In Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pages 2128–2131. IEEE, 2010.
- [Truong 2010] T.T.B. Truong. *Home Automation Monitoring for Assisted Living Services and Healthcare*. PhD thesis, Laboratoire en Sciences et Techniques de l'Information, de la Communication et de la Connaissance, Université de Bretagne-Sud, 2010.
- [United Nations 2009] United Nations. *Department of Economic and Social Affairs - Population Division : World Population Ageing (2009)*, 2009.
- [Waldner 2007] J.B. Waldner. *Nano-informatique et intelligence ambiante : inventer l'ordinateur du XXI siècle*. Recherche, vol. 67, page 02, 2007.
- [Wang et al. 2004] X.H. Wang, D.Q. Zhang, T. Gu et H.K. Pung. *Ontology based context modeling and reasoning using OWL*. In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pages 18–22. Ieee, 2004.



- [Weiser 1991] M. Weiser. *The computer for the 21st century*. Scientific American, vol. 265, no. 3, pages 94–104, 1991.
- [Weiser 1993] M. Weiser. *Some computer science issues in ubiquitous computing*. Communications of the ACM, vol. 36, pages 75–84, 1993.
- [World Health Organization 2011] World Health Organization. *World report on disability*. Geneva : WHO, 2011.
- [Yu et al. 2006] C.R. Yu, C.L. Wu, C.H. Lu et L.C. Fu. *Human localization via multi-cameras and floor sensors in smart home*. In Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on, volume 5, pages 3822–3827. IEEE, 2006.
- [Zamaï et al. 2007] E. Zamaï, F. Rigaud, J.F. Pétin, P. Berruet et A. Toguyeni. *Architectures de pilotage de procédés industriels*. Techniques de l'Ingénieur, 2007.
- [Zhang & MoUuari 2004] D. Zhang et M. MoUuari. «*Aware home understanding life activities*». Toward a humanfriendly assistive environment : ICOST, page 186, 2004.
- [Zhang & Xu 2004] Y. Zhang et B. Xu. *A survey of semantic description frameworks for programming languages*. ACM Sigplan Notices, vol. 39, no. 3, pages 14–30, 2004.
- [Zouba et al. 2009] N. Zouba, F. Brémond, M. Thonnat, A. Anfosso, E. Pascual, P. Mallea, V. Mailland et O. Guerin. *A computer system to monitor older adults at home : preliminary results*. Gerontechnology, vol. 8, no. 3, pages 129–139, 2009.

# Publications personnelles

1. A Non-Intrusive Context-Aware System for Ambient Assisted Living in Smart Home. Allègre, W. and Burger, T. and Antoine, J.Y. and Berruet, P. and Departe, J.P. Submitted in the Health and Technology Journal in 2012.
2. A non-intrusive monitoring system for ambient assisted living service delivery. Allègre, W. and Burger, T. and Berruet, P. and Antoine, J.Y. Proceedings of the 10th International Conference On Smart home and health Telematics (ICOST), Artimino, 2012.
3. Model-Driven Flow for Assistive Home Automation System Design. Allègre, W. and Burger, T. and Berruet, P. Proceedings of the 18th International Federation of Automatic Control (IFAC) world congress, Milano, 2011.
4. Ambient Assisted Living with Linux. Allègre, W. and Séguin, C. and Burger, T. and De Lamotte, F. and Berruet, P. and Philippe, J.L. and Diguët, J.P. Proceedings of the 1st Embed With Linux (EWiLi) workshop, Saint-Malo, 2012.
5. Conception d'un système domotique pour l'assistance aux personnes dépendantes. Allègre, W. and Burger, T. and Berruet, P. and Departe, J.P. Journal des Sciences et Technologies pour le Handicap, 2012.
6. Modèle de supervision d'interactions non-intrusif basé sur les ontologies. Allègre, W. and Burger, T. and Berruet, P. and Antoine, J.Y. and others. Actes du congrès Extraction et Gestion des Connaissances (EGC), Bordeaux, 2012.
7. Aide à la conception d'un système domotique pour l'assistance aux personnes à mobilité réduite. Allègre, W. and Burger, T. and Berruet, P. and others. Actes de la conférence Majestic 2010, Bordeaux, 2012.

## **Vidéos de démonstration :**

1. IntelHome Designer : Model-Driven Flow  
<http://www.youtube.com/watch?v=B4FJcdDBa6Q>
2. IntelHome Designer : Implementation  
<http://www.youtube.com/watch?v=0grL24U5duM>



# Liste des Figures

|      |   |     |
|------|---|-----|
| 1    | Organisation générale du mémoire (décomposition en chapitres) . . . . .   | 7   |
| 1.1  | Évolution de la population française métropolitaine de 2000 à 2050 . . . . .  | 11  |
| 1.2  | Évolution de la population mondiale (> 60 ans) de 1950 à 2050 . . . . .   | 12  |
| 1.3  | Personnes avec problèmes de santé ou déficiences selon l'âge en Europe . . . . .  | 14  |
| 1.4  | Schéma conceptuel du Processus de Production du Handicap . . . . .  | 16  |
| 1.5  | Évolution des systèmes électroniques et des systèmes domotiques . . . . .   | 22  |
| 1.6  | Organisation générale d'un système domotique d'assistance . . . . .   | 26  |
| 1.7  | Installation du système de tracking vidéo (a) et interface utilisateur (b) de l'application " <i>Image-based experience sampling</i> " du MIT . . . . . | 29  |
| 1.8  | Processus de conception centré utilisateur pour la domotique . . . . .  | 35  |
| 1.9  | Schéma de collaboration pour la conception domotique d'assistance . . . . .   | 37  |
| 1.10 | Principe général pour l'analyse sémantique des logs domotiques . . . . .  | 40  |
| 2.1  | Concepts fondamentaux de l'IDM : hiérarchisation des modèles . . . . .  | 46  |
| 2.2  | Illustration des concepts de l'IDM sur un ouvre-porte automatique . . . . .   | 47  |
| 2.3  | Processus de transformation de modèles avec ATL . . . . .   | 48  |
| 2.4  | Composantes fondamentales d'un langage de modélisation . . . . .  | 50  |
| 2.5  | Relations entre système, modèle, méta-modèle et langage . . . . .   | 52  |
| 2.6  | Extrait du méta-modèle <i>Environnement</i> . . . . .   | 60  |
| 2.7  | Composant <i>Porte</i> vu comme une interface entre deux pièces . . . . .   | 61  |
| 2.8  | Représentation graphique du fonctionnement d'une ressource <i>R1</i> . . . . .  | 63  |
| 2.9  | Correspondance entre états, super-états, opérations et fonctions . . . . .  | 64  |
| 2.10 | Extrait du méta-modèle <i>Interaction</i> . . . . .   | 65  |
| 2.11 | Activation d'un mode d'interdiction sur la ressource <i>R1</i> . . . . .  | 68  |
| 2.12 | Représentation graphique de l'activation de deux modes sur <i>R1</i> . . . . .  | 70  |
| 2.13 | Taxonomie d'interactions (services et modes) et domaine de définition . . . . .   | 71  |
| 3.1  | Niveaux de modélisation de la spécification MDA . . . . .   | 77  |
| 3.2  | Flot de conception théorique basé sur une architecture MDA . . . . .  | 79  |
| 3.3  | Architecture de déploiement du middleware Danah . . . . .   | 81  |
| 3.4  | Extrait du méta-modèle <i>Danah</i> . . . . .   | 81  |
| 3.5  | Interface utilisateur Danah affichée sur le PDA . . . . .   | 90  |
| 3.6  | Organisation du logiciel <i>IntelHome Designer</i> en perspectives . . . . .  | 91  |
| 3.7  | Tableau de bord ( <i>Dashboard</i> ) reprenant les étapes de la conception . . . . .  | 92  |
| 4.1  | Utilisation de l'ontologie pour la supervision non-intrusive . . . . .  | 109 |
| 4.2  | Extrait de la taxonomie de l'ontologie (T-Box) . . . . .  | 111 |
| 4.3  | Extrait de l'ontologie pour la localisation de l'utilisateur . . . . .  | 112 |
| 4.4  | Extrait de l'ontologie pour l'intention de l'utilisateur . . . . .  | 113 |
| 4.5  | Extrait de l'ontologie : localisation, intention et proposition de services . . . . .   | 114 |
| 4.6  | Extrait du méta-modèle OWL (taxonomie) . . . . .  | 116 |
| 4.7  | Spécification ODM mise en oeuvre par transformations ATL . . . . .  | 117 |
| 4.8  | Flot de conception pour le contrôle-commande domotique . . . . .  | 118 |

|      |   |     |
|------|---|-----|
| 4.9  | Architecture logicielle pour la supervision basée sur les ontologies . . .  | 120 |
| 5.1  | Photos des appartements tremplins du CMRRF de Kerpape (chambre de l'appartement à gauche, cuisine du studio à droite) . . . . . | 126 |
| 5.2  | Plan des appartements tremplins du CMRRF de Kerpape . . . . .   | 127 |
| 5.3  | Étape 0 : Importation du fichier de configuration . . . . .   | 132 |
| 5.4  | Étape 0 : Importation à partir du tableau de bord . . . . .   | 133 |
| 5.6  | Étape 2 : Transformation ENV→INT à partir du tableau de bord . . . . .  | 133 |
| 5.5  | Étape 1 : Modélisation de l'environnement (appartement T2) . . . . .  | 134 |
| 5.7  | Étape 3 : modélisation des interactions pour l'utilisateur du cas d'étude .   | 135 |
| 5.8  | Étape 4 : Génération du PSM Danah à partir du tableau de bord . . . . .   | 137 |
| 5.9  | Interface utilisateur Danah affichée sur le PDA . . . . .   | 137 |
| 5.10 | Résultats de l'expérimentation - C : conformité (%) . . . . .   | 139 |
| 5.11 | Résultats de l'expérimentation - T : temps (min) . . . . .  | 140 |
| 5.12 | Temps unitaire de modélisation d'interactions . . . . .   | 140 |
| 5.13 | Plan de l'appartement Domus tiré de [Domus] . . . . .   | 144 |
| 5.14 | Application pour le suivi de personnes dans l'appartement Domus . . .   | 147 |
| B.1  | Importation du fichier de configuration KNX dans ETS 3 . . . . .  | 180 |

# Liste des Tableaux

|     |   |     |
|-----|---|-----|
| 2.1 | Langages de modélisation pour la conception domotique . . . . .   | 55  |
| 2.2 | Vues Environnement et Interactions et concepts associés . . . . .                                       | 62  |
| 3.1 | Syntaxe graphique pour le modèle d'environnement <i>MMEnvironment</i> . .                               | 94  |
| 3.2 | Syntaxe graphique pour le modèle d'interactions <i>MMInteractions</i> . . . . .                         | 96  |
| 4.1 | Extraits d'une base de connaissances : T-Box à gauche / A-Box à droite .                                | 103 |
| 4.2 | Règles SWRL pour la localisation de l'utilisateur . . . . .   | 112 |
| 4.3 | Règle SWRL pour l'intention de l'utilisateur . . . . .  | 113 |
| 4.4 | Règles SWRL pour la proposition de services . . . . .   | 114 |
| 4.5 | Correspondances entre les concepts UML/OWL . . . . .  | 117 |
| 5.1 | Nombre d'effecteurs (actionneurs et capteurs) selon l'emplacement . . .                                 | 129 |
| 5.2 | Hypothèses testées au cours des expérimentations <i>EXP<sub>1</sub></i> et <i>EXP<sub>2</sub></i> . . . | 131 |
| 5.3 | Exemple de logs issus des jeux de données Domus . . . . .   | 145 |
| 5.4 | Jeu de données de la série 1 / utilisateur 1 / jour 2 (extrait retenu) . . .                            | 146 |
| 5.5 | Résultats de l'application de supervision sur les jeux de données Domus                                 | 149 |





## Installation du logiciel

Cette annexe liste les étapes pour installer et configurer le logiciel IntelHome Designer (version 6.0) : du téléchargement d'*Obeo Designer* jusqu'à la mise en place de l'environnement logiciel. Cette procédure a été testée sous *Ubuntu 11.10*, *Ubuntu 12.04*, *Windows Vista* et *Windows Seven*.

### Téléchargement d'*Obeo Designer*

1. À partir du site <http://www.obeodesigner.com/download>, télécharger la version architecte (licence académique ou version d'évaluation valable 1 mois)
2. Extraire le dossier compressé téléchargé sur un PC

### Lancement d'*Obeo Designer*

1. Dans le dossier extrait, créer un dossier *workspace-IntelHomeDesigner*
2. Lancer *Obeo Designer* en sélectionnant ce nouveau dossier (workspace courant)
3. Dans la vue *Model Explorer*, clic-droit → *Import...* → *Existing Projects into Workspace*
4. Sélectionner l'archive *workspace-IntelHomeDesigner.zip* (copie dans le workspace)

### Lancement de l'application *IntelHome Designer*

1. Dans la vue *Model Explorer*, clic-droit sur le dossier *labsticc.intelhomedesigner* → *Run As...* → *Run Configurations...*
2. Créer une nouvelle configuration Eclipse avec le nom "*IntelHomeDesigner*", l'emplacement "*workspace\_loc/./runtime-IntelHomeDesigner*" et les Arguments : "*XX:MaxPermSize=512m*" pour limiter la JVM Java.
3. Appliquer la configuration et lancer l'application IntelHome Designer
4. De la même manière que pour le workspace, importer le dossier compressé "*runtime-IntelHomeDesigner.zip*"



## Configuration de l'application *IntelHome Designer*

### Configuration des transformations ATL & Acceleo

Pour chaque dossier ATL (de la vue *Model Explorer*) qui comprend les fichiers de transformation ATL, il faut définir une configuration pour préciser les modèles d'entrée et les modèles de sortie (cf flot de conception dirigé par les modèles) : faire un clic-droit sur le dossier ATL → *Run As...* → *Run Configurations...*, puis créer des configurations ATL avec les paramètres suivants :

1. *env2env\_configuration* :
  - ATL Module = `"/labsticc.env2env/env2env.atl"`
  - *MMEnvironment* = `"/labsticc.intelhomedesigner.runtime/metamodel/environment.ecore"`
  - *Source Models IN* = `"/labsticc.intelhomedesigner.runtime/model_env/My.environment"`
  - *Target Models OUT* = `"/labsticc.intelhomedesigner.runtime/model_env/My.environment"`
  - Dans l'onglet "Advanced", cocher "Allow inter-model references"
2. *env2int\_configuration* :
  - ATL Module = `"/labsticc.env2int/env2int.atl"`
  - *MMEnvironment* = `"/labsticc.intelhomedesigner.runtime/metamodel/environment.ecore"`
  - *MMInteraction* = `"/labsticc.intelhomedesigner.runtime/metamodel/interaction.ecore"`
  - *Source Models IN* = `"/labsticc.intelhomedesigner.runtime/model_env/My.environment"`
  - *Target Models OUT* = `"/labsticc.intelhomedesigner.runtime/model_int/My.interaction"`
3. *intel2danah\_configuration* :
  - ATL Module = `"/labsticc.intel2danah/intel2danah.atl"`
  - *MMEnvironment* = `"/labsticc.intelhomedesigner.runtime/metamodel/environment.ecore"`
  - *MMInteraction* = `"/labsticc.intelhomedesigner.runtime/metamodel/interaction.ecore"`
  - *MMDanah* = `"/labsticc.intelhomedesigner.runtime/metamodel/danah.ecore"`
  - *Source Models IN* = `"/labsticc.intelhomedesigner.runtime/model_env/My.environment"`
  - *Source Models IN1* = `"/labsticc.intelhomedesigner.runtime/model_int/My.interaction"`
  - *Target Models OUT* = `"/labsticc.intelhomedesigner.runtime/model_danah/My.danah"`

Pour la transformation *Model To Text Acceleo*, de la même manière que pour les transformations ATL, créer une configuration avec les paramètres suivants :

1. *danah2text\_configuration* :
  - *Project* = `"labsticc.danah2text"`
  - *Main class* = `"labsticc.danah2text.files.Danah2text"`
  - *Model* = `"/labsticc.danah2text/src/labsticc/danah2text/files/danah2text.mtl"`
  - *Target* = `"/labsticc.intelhomedesigner.runtime/configuration_out"`

### Préparation de l'application et de ses vues

1. Importer la vue *DashBoard* (tableau de bord) à partir du menu *Window* → *Show\_View* → *Other...*
2. À partir de la vue *Model Explorer*, afficher ou créer les sessions graphiques `"/labsticc.intelhomedesigner.runtime/model_env/environment.aird"` et `"/labsticc.intelhomedesigner.runtime/model_int/interaction.aird"` puis relancer l'application avant de commencer la modélisation (utilisation du tableau de bord).

Lors de l'import d'un fichier de configuration, les deux modèles graphiques se ferment : l'application peut démarrer.



# B

## Exportation de fichiers

Pour importer les fichiers de configuration dans le framework *IntelHome Designer*, il faut au préalable exporter ces fichiers depuis d'autres logiciels. Cette annexe liste les grandes étapes<sup>1</sup> pour :

1. Exporter des fichiers de configuration *KNX* à partir du logiciel *ETS*
2. Exporter des fichiers de configuration *LIRC* à partir d'une console sous linux

### Fichier de configuration ETS

Dans ETS [KNX Association 1993], il faut activer la vue "*Adresses de groupe*". Comme le montre la figure B, il faut dans un premier temps sélectionner "*Exportation d'Adresses de Groupe...*" (clic droit sur le label *Groupes Principaux*) pour générer un fichier texte sans en-tête (case "*Exporter avec en-tête*" décochée sur la deuxième fenêtre) avec le format "*1/1 (Nom/Adr.)*" où les adresses seront séparées par des points-virgules (case "*point-virgule*" cochée).

### Fichier de configuration LIRC

Sous Linux, le fichier de configuration LIRC liste les commandes infrarouges pour les retransmettre à l'aide d'un émetteur infrarouge (e.g. *IguanaIR*). Ces fichiers de configuration peuvent s'exporter depuis "*http://lirc.sourceforge.net/remotes/*" dans le cas où il existe un modèle de l'équipement que l'on cherche à contrôler. Dans le cas où les codes infrarouges ont été appris et enregistrés sur un système Linux avec LIRC, le fichier de configuration "*lircd.conf*" se retrouve par défaut dans "*/etc/lirc/*".

---

1. Pour plus d'informations sur les procédures à suivre sur l'édition de ces fichiers de configuration, se reporter aux guides utilisateur des programmes *ETS* et *LIRC*.

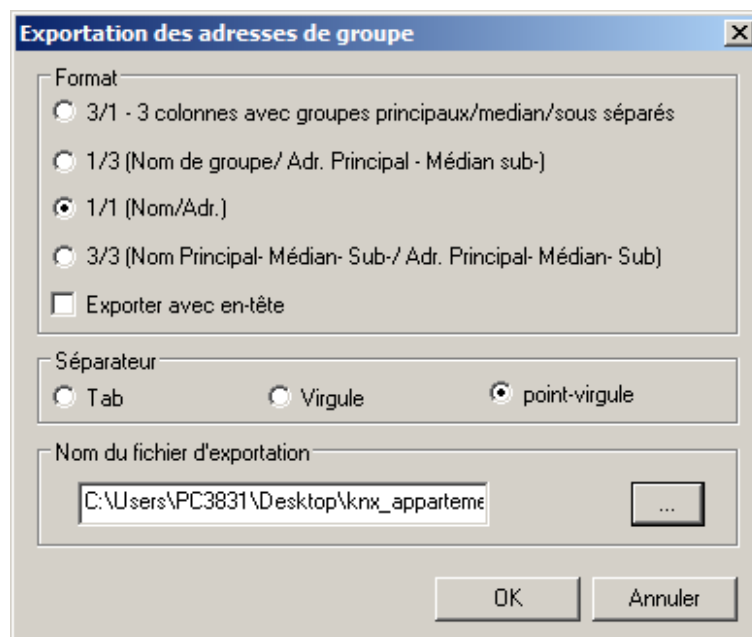
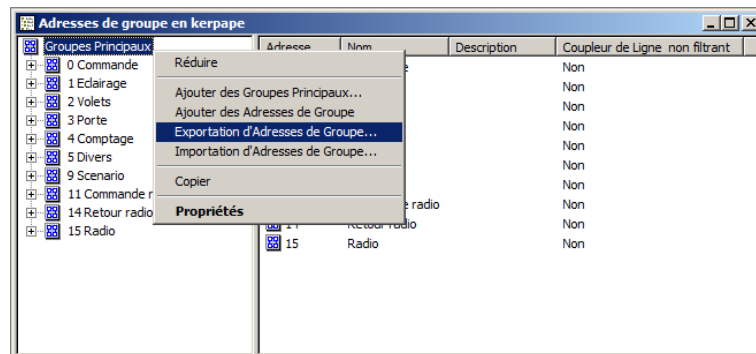


FIGURE B.1 – Importation du fichier de configuration KNX dans ETS 3



# Questionnaire

Échelle :  oui, très  assez  peu  non, pas du tout

## À propos de vous

1. Votre nom : \_\_\_\_\_
2. Dans votre activité professionnelle, avez-vous déjà ressenti le besoin de configurer un système domotique ?  
 oui  non
3. Pensez-vous être capable de le faire avec ce logiciel ?  
 oui  non

## À propos de l'exercice à réaliser

4. Le travail demandé vous a-t-il paru difficile dans l'ensemble ?  
oui, très ——— non, pas du tout
5. Les instructions fournies sont-elles assez claires ?  
oui, très ——— non, pas du tout
6. Les instructions fournies sont-elles suffisantes pour réaliser ce travail ?  
oui, très ——— non, pas du tout

## À propos de l'activité de modélisation

7. Estimez-vous que l'une des parties est plus complexe à modéliser ?  
 oui, ENV  oui, INT  non  
 Si oui, pourquoi ? \_\_\_\_\_
8. Quelle partie avez vous préféré réaliser ?  
 ENV  INT  pas de préférences

9. **Estimez-vous que la partie ENV était difficile à modéliser ?**  
oui, très ——— non, pas du tout
10. **Estimez-vous que la partie INT était difficile à modéliser ?**  
oui, très ——— non, pas du tout
11. **Qu'est ce qui vous a posé le plus de difficultés pour la modélisation d'interactions ?**  
 la compréhension de la description textuelle  la traduction à l'aide de l'outil  
 pourquoi ? \_\_\_\_\_

## À propos du logiciel

12. **Avez-vous trouvé le logiciel IntelHome Designer complexe à utiliser ?**  
oui, très ——— non, pas du tout
13. **Quel moyen de création d'éléments graphiques avez-vous préféré ?**  
 menu contextuel (souris)  palette graphique  
 pourquoi ? \_\_\_\_\_
14. **Les éléments graphiques de la vue environnement sont-ils facilement compréhensibles ?**  
oui, très ——— non, pas du tout
15. **Les éléments graphiques de la vue interaction sont-ils facilement compréhensibles ?**  
oui, très ——— non, pas du tout
16. **Construire un schéma graphique pour définir les scénarios vous semble t-il intuitif ?**  
oui, très ——— non, pas du tout

## Commentaires / Remarques

17. **Décrivez ici vos impressions, remarques ou commentaires libres concernant les points précédemment abordés :**

---



---



---



---



---

18. **Que pensez-vous du temps d'apprentissage passé sur l'exemple illustratif ? Est-il suffisant pour prendre en main le logiciel ?**

---



---



---



---



---



## Consentement éclairé

### Présentation du logiciel

Il s'agit de la première version d'*IntelHome Designer*, actuellement en cours de développement. Ce logiciel vise à rendre la conception de systèmes domotiques<sup>1</sup> (actuellement réservée à des techniciens/domoticiens) accessible à des personnes non expertes de ce domaine. Que ce soit un ergothérapeute (cadre institutionnel) ou bien un membre de la famille (cadre du maintien à domicile), nous pensons que cette personne est la mieux placée pour prendre en compte les besoins de vie exprimés par l'utilisateur en situation de handicap.

### Conditions de l'expérimentation

L'objectif de cette expérimentation est d'avoir un premier retour sur l'utilisation du logiciel par une personne non-experte en informatique et électronique. Nos observations nous permettront d'évaluer le logiciel et de l'améliorer pour qu'il puisse être utilisé en situation réelle. Au cours de cette expérimentation, il sera demandé de suivre un tutoriel pour vous guider dans les différentes étapes de la conception (définition pour un cas d'étude des équipements domotiques, des pièces, des services domotiques). Différentes métriques seront relevées (temps, mouvements de souris...). Il est important de savoir que **ce n'est pas vous que nous testons**, mais bien le logiciel de configuration. À ce titre, les données personnelles qui seront enregistrées pendant cette expérimentation resteront confidentielles. Aucun profit d'ordre financier ne sera par ailleurs fait des données enregistrées ou de leur exploitation. Ces dernières serviront au laboratoire Lab-STICC dans le cadre de travaux de recherche.

Ce test devrait prendre environ **deux heures**, mais vous pourrez l'arrêter quand vous le voudrez si vous le souhaitez. Une **place de cinéma** (cinéville) sera offerte à tous les participants.

### Consentement éclairé

Je suis conscient et j'accepte les conditions précitées dans lesquelles se dérouleront l'expérience.

Nom et signature :

---

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.







## Tutoriels

Deux tutoriels sont présentés :

1. Le tutoriel utilisé pour l'expérimentation auprès d'étudiants
2. Le tutoriel utilisé pour l'expérimentation auprès d'ergothérapeutes

---

# IntelHome Designer version 1.2

Kerpape MFRRC Smart Home

---

**Étude expérimentale pour l'évaluation de l'outil de CAO**

**(Groupe A)**

Willy Allègre (Lab-STICC / Université de Bretagne Sud)

---

# Introduction

## Présentation d'*IntelHome Designer*

Il s'agit de la première version d'*IntelHome Designer*, actuellement en cours de développement. Cet outil vise à rendre la conception de systèmes domotiques<sup>1</sup> (actuellement réservée à des techniciens/domoticiens) accessible à des personnes non expertes de ce domaine. Que ce soit un ergothérapeute (cadre institutionnel) ou bien un membre de la famille (cadre du maintien à domicile), cette personne est avant tout capable de prendre en compte les besoins de vie exprimés par l'utilisateur en situation de handicap.

## Présentation du cas d'étude

Nous nous intéressons aux appartements tremplins du CMRRF<sup>2</sup> de Kerpape, conçu pour évaluer l'aptitude au retour à domicile de personnes en situation de handicap. Entièrement équipés en domotique, ces appartements sont un cadre idéal pour tester l'autonomie des patients en situation réelle.

Pour ce cas d'étude, nous considérons deux patients qui occupent ces appartements dans le cadre d'un court séjour :

- L'*utilisateur 1* occupe l'*appartement 1* (studio de 35m<sup>2</sup>), équipé d'un bus domotique propriétaire<sup>3</sup>.
- L'*utilisateur 2* occupe l'*appartement 2* (T2 de 75m<sup>2</sup>), équipé d'un bus domotique ouvert<sup>4</sup>.

Le deuxième appartement possède une pièce de vie qui peut être partagée par les deux patients pour éventuellement y accueillir leurs familles respectives (cf figure 6 de l'annexe B).

## Objectif du tutoriel

Ce tutoriel propose de concevoir, à l'aide de l'outil de conception *IntelHome Designer*, le code de commande d'un système domotique adapté aux besoins des patients présents dans les appartements tremplins.

À ce stade, l'installation et l'intégration des solutions domotiques (équipements, câblage, etc.) ont déjà été réalisées par un domoticien. Il faut donc à présent se mettre dans la peau du concepteur non-expert, chargé de définir un environnement de vie adapté aux besoins des utilisateurs présentés dans la section précédente.

Pour cela, différentes activités de modélisation sont proposées<sup>5</sup> (cf figure 1) :

1. *ENV* : la modélisation de l'environnement domotique et ses équipements (pour l'*appartement 1*)
2. *INT2* : la modélisation des interactions répondant aux besoins de l'*utilisateur 2* (de l'*appartement 2*)
3. *INT1* : la modélisation des interactions répondant aux besoins de l'*utilisateur 1* (de l'*appartement 1*)

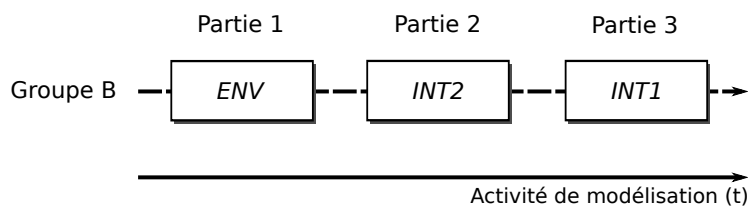


FIGURE 1 – Activités de modélisation en fonction du temps (groupe B)

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. Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelle de Kerpape : <http://www.kerpape.mutualite56.fr/>

3. Legrand : <http://www.legrand.fr/>

4. KNX : <http://www.knx.org/>

5. Ce tutoriel propose de suivre la méthode de modélisation et d'utiliser un langage graphique de description de systèmes domotiques proposés dans "*Model-Driven Flow for Home Automation System Design*", W. Allègre, T. Burger, P. Berruet.

Les activités de modélisation à réaliser sont décomposés en plusieurs questions représentées par **Question**  
 Les interactions avec la personne chargée de superviser l'expérimentation sont signalés par **Interaction**

**Question 1** : Prendre connaissance de l'organisation générale de l'outil *IntelHome Designer* présenté dans l'annexe A.

## 1 Modélisation de l'environnement domotique

### 1.1 Intégration des composants

Le premier travail du concepteur non-expert consiste à modéliser l'environnement domotique dans lequel se trouvent les utilisateurs. Cette modélisation se base sur la notion de *composants* : un environnement domotique est vu comme un ensemble de composants imbriqués les uns dans les autres. Ainsi, les composants *D1*, *D11*, *D21* et *L1* (équipements domotiques) peuvent être assemblés pour former un composant de plus haut niveau d'abstraction : *Entrance* (cf plan de la figure 6 de l'annexe B).

Dans ce cas d'étude, en plus des équipements domotiques, on ne considère que des composants à l'échelle d'une pièce :

|                     | Lumière   | Porte électrique  | Volets électriques   | Lit                    | Télévision               | Musique                  | Téléphone               |
|---------------------|---|---|--|------------------------|--------------------------|--------------------------|-------------------------|
| <i>Entrance</i>     | <i>L1</i> <sup>1</sup>                          | <i>D1</i> <sup>1</sup> <i>D11</i> <sup>2</sup> <i>D21</i> <sup>1</sup>  |  |                        |                          |                          |                         |
| <i>Kitchen1</i>     | <i>L11</i> <sup>2</sup>                         | <i>D11</i> <sup>2</sup>   | <i>SH11</i> <sup>2</sup>   |                        |                          |                          |                         |
| <i>BedRoom1</i>     | <i>L12</i> <sup>2</sup>                         | <i>D12</i> <sup>2</sup>   | <i>SH12</i> <sup>2</sup>   | <i>B1</i> <sup>3</sup> | <i>TV1</i> <sup>3</sup>  |                          |                         |
| <i>BathRoom1</i>    | <i>L13</i> <sup>2</sup>                         | <i>D12</i> <sup>2</sup>   |  |                        |                          |                          |                         |
| <i>LivingRoom</i>   | <i>L21</i> <sup>1</sup> <i>L22</i> <sup>1</sup> | <i>D21</i> <sup>1</sup> <i>D22</i> <sup>1</sup>                         | <i>SH21</i> <sup>1</sup> <i>SH22</i> <sup>1</sup> <i>SH23</i> <sup>1</sup> |                        | <i>TV21</i> <sup>3</sup> | <i>HF21</i> <sup>3</sup> | <i>PH1</i> <sup>3</sup> |
| <i>BedRoom2</i>     | <i>L23</i> <sup>1</sup>                         | <i>D22</i> <sup>1</sup> <i>D23</i> <sup>1</sup> <i>D24</i> <sup>1</sup> | <i>SH24</i> <sup>1</sup> <i>SH25</i> <sup>1</sup>                          | <i>B2</i> <sup>3</sup> | <i>TV22</i> <sup>3</sup> |                          |                         |
| <i>BathRoom2</i>    | <i>L24</i> <sup>1</sup>                         | <i>D23</i> <sup>1</sup>   | <i>SH26</i> <sup>1</sup>   |                        |                          |                          |                         |
| <i>StorageRoom2</i> | <i>L25</i> <sup>1</sup>                         | <i>D24</i> <sup>1</sup>   | <i>SH27</i> <sup>1</sup>   |                        |                          |                          |                         |

TABLE 1 – Décomposition en composants (équipements et pièces) des appartements tremplins du CMRRF (protocole de communication associé aux équipements domotiques : <sup>1</sup> : *KNX*, <sup>2</sup> : *Legrand*, <sup>3</sup> : *infrarouge*)

**Interaction 1** : Signaler que l'activité de modélisation *ENV* peut commencer.

**Question 2** : À partir de la figure 6 (annexe B), définir et nommer les pièces de l'appartement 1.

**Question 3** : À partir de l'annexe B et du tableau 1, définir et nommer les équipements domotiques présents dans les pièces de l'appartement 1. Les portes étant vus comme des interfaces entre deux pièces, définir pour chaque porte deux liens (*Door link*) vers des pièces.

À ce stade, toutes les pièces et tous les équipements listés dans le tableau 1 sont modélisés. Pour permettre de contrôler ces derniers, différents "organes de commandes" doivent être modélisés, puis configurés.

### 1.2 Intégration et configuration des bornes domotiques

Les bornes domotiques sont responsables de la prise en charge des différents protocoles de communication supportés par les équipements. Dans ce cas d'étude, une borne est chargée de commander les bus domotiques *HAB\_KNX* et *HAB\_LGD* présents dans le hall d'entrée. Ces derniers permettent de contrôler respectivement les équipements *KNX* et les équipements *Legrand*.

**Question 4** : Dans le hall d'entrée (*Entrance*), définir le bus domotique *HAB\_LGD* pour contrôler les équipements de l'appartement 1. Définir les paramètres suivants :

- son nom
- le protocole de communication pris en charge (*Legrand*)
- les liens vers les équipements *Legrand* (portant l'indice <sup>2</sup> dans le tableau 1)

**Question 5** : Intégrer une borne domotique dans le hall d'entrée pour contrôler les deux bus domotiques (*HAB\_KNX* et *HAB\_LGD*). Pour la configuration de la borne, définir les paramètres suivants :

- son nom : *Server1*
- les protocoles de communication pris en charge (*KNX* et *Legrand*)
- un lien vers le bus *HAB\_KNX*
- un lien vers le bus *HAB\_LGD*

**Question 6** : Intégrer et configurer une autre borne domotique pour contrôler en infrarouge les équipements multimédia de l'appartement 1.

À ce stade, tous les équipements sont connectés à une borne domotique : soit directement, soit par l'intermédiaire d'un bus. Les équipements infrarouges sont directement connectés à des bornes domotiques. Les autres équipements sont connectés à la borne *Server1* par l'intermédiaire des bus domotiques. Les équipements de l'appartement 1 sont connectés au bus domotique *Legrand* (*HAB\_LGD*), lui-même connecté à la borne domotique *Server1*, placée dans le hall d'entrée. Cette borne est également connectée au bus domotique *KNX* (*HAB\_KNX*) pour commander les équipements de l'appartement 2.

### Configuration des commandes domotiques

La programmation des centrales domotiques a déjà été réalisée par un domoticien. Il faut à présent faire le lien entre les commandes enregistrées par ce technicien et les opérations des équipements modélisés.

**Question 7** : Pour tous les équipements domotiques (composants élémentaires) de l'appartement 1, renseigner un lien IO Label qui porte le même nom que le composant (vue *propriétés*).

**Remarque 1** : Le protocole de communication *Legrand* est une technologie propriétaire : il n'est pas possible de s'interfacer directement avec la centrale. Un module infrarouge est alors intégré dans la centrale pour récupérer les commandes infrarouges émises par la borne *Server1* et les interpréter. À partir de ces commandes infrarouges, la centrale domotique commande ensuite les équipements de l'appartement 2.

**Remarque 2** : L'association entre commande et opération dépasse le cadre de ce tutoriel. De la même manière que dans la question 9, un lien vers une commande *infrarouge* ou *KNX* doit être renseigné.

**Interaction 2** : Sauvegarder le modèle d'environnement et indiquer que cette partie est terminée.

À ce stade, les appartements tremplins du centre de Kerpape ont été modélisés. Il s'agit de la *partie physique* de la conception, qu'il faut à présent adapter pour répondre aux besoins particuliers des utilisateurs de ce cas d'étude (*partie logique*). Un modèle d'interactions doit donc être défini pour chaque utilisateur.

## 2 Modélisation des interactions utilisateur-système

À partir des équipements du modèle d'environnement et de leurs opérations, un modèle minimaliste d'interactions est défini. Ce modèle comprend des services élémentaires soumis à une condition d'activation manuelle (ex : l'utilisateur demande d'allumer la lumière L11) et des états qu'il sera possible d'interdire par l'intermédiaire des modes (ex : on interdit l'état allumé de la lumière L1 dans la journée). Le deuxième travail du concepteur non-expert consiste donc à enrichir ce modèle d'interactions pour répondre aux besoins des utilisateurs.

**Question 8** : Prendre connaissance des objets graphiques utiles pour modéliser des interactions (services et modes) présentée dans l'annexe D.

**Interaction 3** : Signaler que l'activité de modélisation *INT1* peut commencer.

### 2.1 Modélisation des interactions de l'utilisateur 1

Dans un premier temps, l'utilisateur a besoin d'automatisation pour compenser son handicap physique. L'ergothérapeute, accompagné de l'équipe médicale et paramédicale, définit des scénarios de vie utiles dans la perspective du retour à domicile.

Le scénario "*Urgence*" permet par exemple d'effectuer la séquence suivante :

1. activer la touche 2 du téléphone PH1 pour composer un numéro d'urgence programmé
2. ouvrir la porte d'entrée des appartements tremplins (*D1*)
3. ouvrir la porte d'entrée de l'appartement 2 (*D11*), pour faciliter l'accès à une tierce personne

**Question 9** : Définir le service "*Urgence*" de type scénario soumis à une condition d'activation manuelle, sans prise en compte de pannes.

\* \* \*

Pour prévenir des éventuels pannes des actioneurs, des scénarios avec prise en compte du résultat d'exécution sont définis.

Par exemple le scénario "*Allumer Chambre 1*" permet d'effectuer la séquence suivante :

1. allumer la lumière de la chambre 1 (*L12*)
2. **si** la lumière *L12* s'est correctement activée → fin du scénario
3. **sinon** allumer la lumière de la pièce d'à côté (*L11*)

**Question 10** : Définir le scénario "*Allumer Chambre 1*" soumis à une condition d'activation manuelle, avec prise en compte de pannes.

\* \* \*

Afin d'automatiser d'avantage le fonctionnement de l'habitat, des services sont soumis à des conditions d'activation automatiques.

Par exemple, le service élémentaire "*Lumière auto*" permet :

1. dans le cas où les volets *SH12* viennent d'être descendus :
2. allumer la lumière de la chambre 1 (*L12*)

**Question 11** : Définir le service élémentaire "*Lumière auto*" soumis à une condition d'activation automatique.

\* \* \*

Certains services ne sont pas assez réguliers pour être entièrement automatisés. Dans ce cas, l'avis de l'utilisateur doit être sollicité. Des services avec demande de confirmation sont dans ce cas définis.

Par exemple, le scénario "*Rentrer chambre*" permet :

1. lorsqu'il est 22h, de demander à l'utilisateur situé dans la pièce commune (*LivingRoom*) :
2. ouvrir la porte du hall d'entrée (*D21*)
3. ouvrir la porte de la chambre 2 (*D11*)
4. après une minute d'attente, fermer la porte *D11*

**Question 12** : Définir le scénario "*Rentrer chambre*" soumis à une condition avec demande de confirmation.

\* \* \*

De manière complémentaire aux services, il est possible de définir des interdictions sur l'environnement domotique. La définition de modes d'interdiction permet ainsi de restreindre momentanément l'accès aux services de la pièce commune (*LivingRoom*).

Un mode "*Silence Salon*" est par exemple défini, interdisant l'utilisation des équipements multimédia :

1. entre 22h et 8h :
2. interdire l'état allumé de la télévision *TV21*.
3. interdire l'état allumé du lecteur de musique *HF21*.

**Question 13** : Définir le mode d'interdiction "*Silence Salon*" soumis à une condition d'activation automatique

**Interaction 4** : Sauvegarder le modèle d'interactions 1 et indiquer que cette partie est terminée.

## 2.2 Modélisation des interactions de l'utilisateur 2

L'utilisateur 2 présente des troubles cognitifs débutant dans le cadre d'une démence de type Alzheimer. Pour l'instant, il s'agit surtout de légers troubles mnésiques (petits oublis). Progressivement, ces troubles de mémoire vont s'accroître, entraînant une désorientation temporo-spatiale, et plus tard des fugues. L'environnement domotique doit être capable de s'adapter pour répondre aux besoins émergents.

Dans un premier temps, des scénarios soumis à une condition d'activation manuelle sont définis pour automatiser le fonctionnement de l'appartement 2.

Le scénario "*Je dors*" permet par exemple d'effectuer la séquence suivante :



1. baisser le lit électrique de la chambre 2 (*B2*)
2. éteindre la télévision de la chambre 2 (*TV22*)
3. après une minute d'attente, éteindre la lumière *L23*

**Question 14** : Définir le service “*Je dors*” de type `scénario` soumis à une condition d'activation manuelle, sans prise en compte de pannes.

\* \* \*

Pour prévenir des éventuels pannes des actioneurs, des scénarios avec prise en compte du résultat d'exécution sont définis dans le salon partagé par les deux utilisateurs.

Par exemple le scénario “*Allumer Salon*” permet d'effectuer la séquence suivante :

1. allumer la lumière principale du salon : *L21*
2. **si** cette lumière s'est correctement activée → fin du scénario
3. **sinon** allumer la lumière secondaire du salon (*L22*)

**Question 15** : Définir le scénario “*Allumer Salon*” soumis à une condition d'activation manuelle, avec prise en compte de pannes.

\* \* \*

Afin d'automatiser d'avantage le fonctionnement de l'habitat, des services sont soumis à des conditions d'activation automatiques.

Par exemple, le scénario “*Silence*” permet :

1. dans le cas où le téléphone *PH1* vient d'être décroché :
2. activer la fonction silence de la télévision *TV21*
3. activer la fonction silence du lecteur de musique *HF21*

**Question 16** : Définir le scénario “*Silence*” soumis à une condition d'activation automatique.

\* \* \*

Pour palier les légers troubles mnésiques de l'utilisateur 2, des services qui nécessitent une **demande de confirmation** sont définis. Ces services, rythmant la journée, aide ce dernier à se repérer dans le temps et permet de stimuler sa mémoire.

Par exemple, le service élémentaire “*Serie TV*” permet :

1. lorsqu'il est 14h :
2. demander à l'utilisateur 2 de mettre la chaîne 1 de la télévision du salon (*TV21*)

**Question 17** : Définir le service élémentaire “*Serie TV*” soumis à une condition avec demande de confirmation.

\* \* \*

La démence de type Alzheimer est à présent au stade sévère. De manière complémentaire aux services et en accord avec l'équipe médicale, l'ergothérapeute définit un certain nombre d'interdictions sur l'environnement domotique dans le but de prévenir d'une éventuelle fugue et assurer la sécurité de ce dernier.

Un mode “*Anti fugue*” est par exemple défini, interdisant l'ouverture de la porte d'entrée pendant la nuit :

1. entre 23h et 8h :
2. interdire l'état ouvert de la porte d'entrée des appartements tremplins (*D1*).

**Question 18** : Définir le mode d'interdiction “*Anti fugue*” soumis à une condition d'activation automatique.

**Remarque** : Au stade sévère de la démence d'Alzheimer, on va aussi privilégier l'automatisation des services qui étaient précédemment activables manuellement ou avec demande de confirmation.

**Interaction 5** : Sauvegarder le modèle d'interactions 2 et indiquer que cette partie est terminée.

\* \* \*

À partir du modèle représentant l'environnement domotique et le modèle d'interactions propre à chaque utilisateur, la configuration pour l'application domotique Danah<sup>6</sup> est automatiquement générée. Chaque utilisateur peut ainsi utiliser cette application pour bénéficier des interactions modélisées dans ce tutoriel.

**Interaction 6** : Remplir le questionnaire sur l'expérimentation *IntelHome Designer*.

---

6. Danah est un logiciel développé au laboratoire Lab-STICC, dédié aux systèmes domotiques d'assistance, qui permet de contrôler son environnement domotique à partir d'une application installée sur un PDA<sup>7</sup>.

## A Présentation du framework

Le framework *IntelHome Designer* est composé de différentes vues (cf **figure 2**) :

1. Le **canevas** permet de manipuler les objets graphiques d'un modèle (environnement ou interaction)
2. La **tablette** permet de définir des objets graphiques pour les mettre sur le canevas
3. La vue **Outline** permet de visualiser globalement le modèle courant (environnement ou interaction)
4. La vue **Properties** permet de définir, pour chaque objet sélectionné, ses propriétés

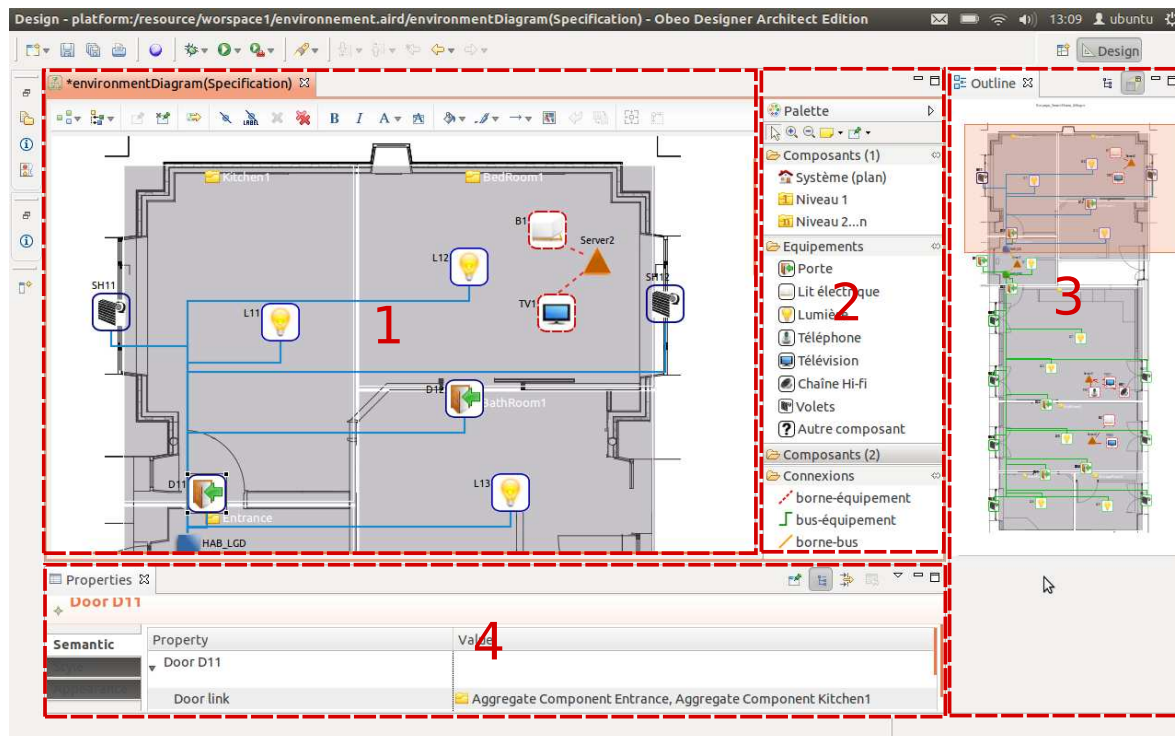


FIGURE 2 – Organisation générale du framework IntelHome Designer

Au dessus du canevas, des filtres et des calques peuvent être ajoutés/retirés pour cacher certains objets graphiques du canevas ou de la palette :

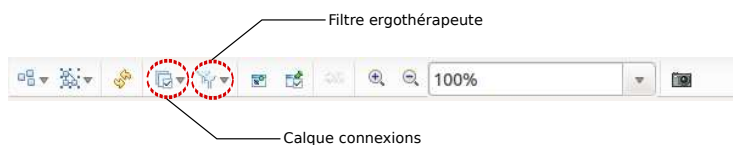


FIGURE 3 – Menu pour la modélisation de l'environnement (filtre / calque)

## B Plan des appartements tremplins

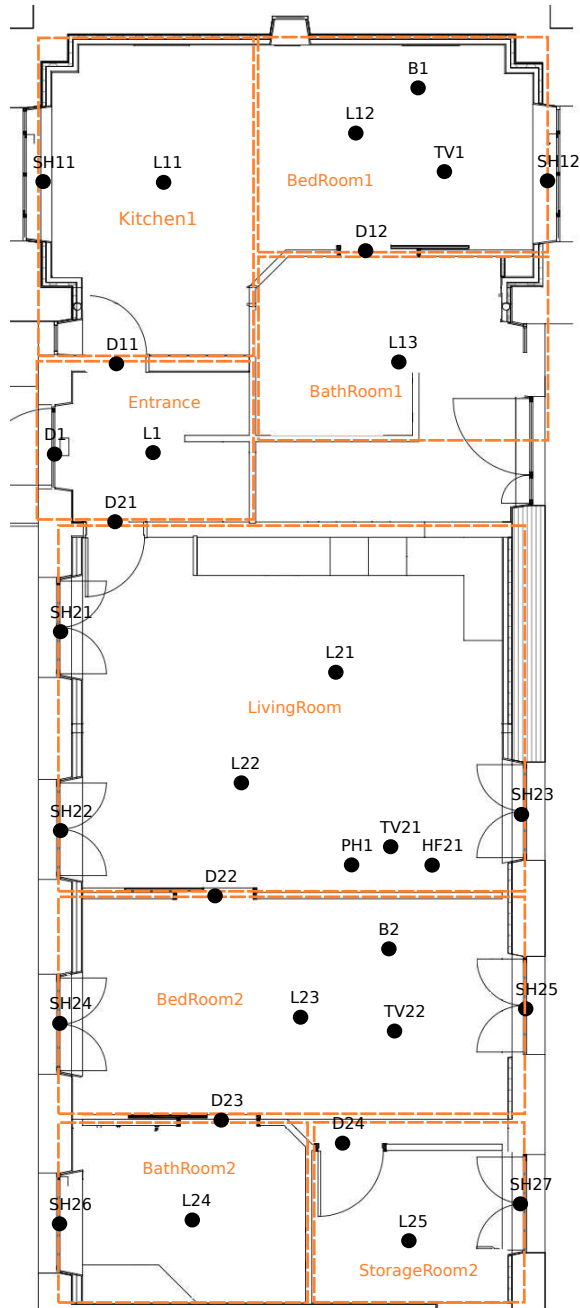
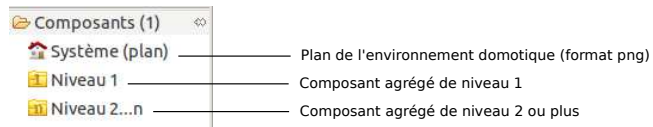


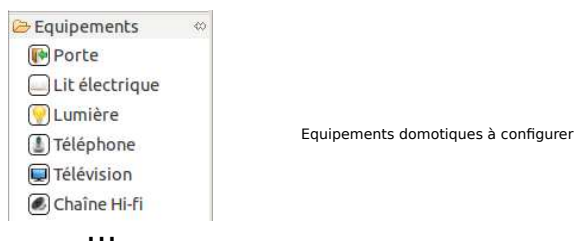
FIGURE 4 – Plan des appartements tremplins du CMRRF de Kerpape

## C Objets graphiques pour la modélisation de l'environnement

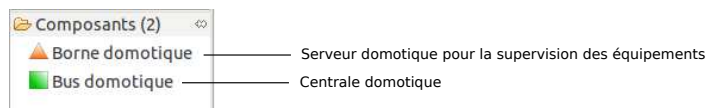
### COMPOSANTS AGREGES



### COMPOSANTS ELEMENTAIRES



### ORGANES DE COMMANDE



### CONNEXIONS (ORGANES DE COMMANDE)

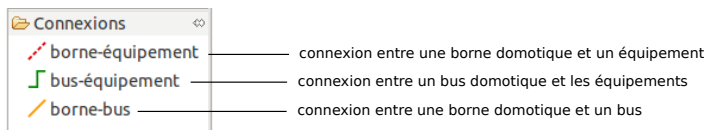


FIGURE 5 – Objets graphiques présents dans la palette de la vue environnement

# D Objets graphiques pour la modélisation d'interactions

## TYPE D'INTERACTIONS

|   |   |   |
|---|---|---|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 1.2em;">Interactions</span> </div>  |   | <p><b>Interaction = Condition d'activation + Action</b></p> |
| <ul style="list-style-type: none"> <li><span style="color: blue;">👤</span> Profil</li> <li><span style="color: green;">📄</span> Service</li> <li><span style="color: red;">📄</span> Mode</li> </ul> | <ul style="list-style-type: none"> <li>Profil lié à un utilisateur</li> <li>Service (permet d'exprimer la formule "je veux")</li> <li>Mode (permet d'exprimer la formule "j'interdis")</li> </ul> |   |

## TYPE DE CONDITIONS D'ACTIVATION

|   |  |
|---|--|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 1.2em;">Type d'activations</span> </div>  |  |
| <ul style="list-style-type: none"> <li><span style="color: blue;">👤</span> Activation manuelle</li> <li><span style="color: grey;">?</span> Demande confirmation</li> <li><span style="color: grey;">🔍</span> Activation automatique</li> </ul> | <ul style="list-style-type: none"> <li>Activation sous demande utilisateur</li> <li>Activation automatique avec demande de confirmation</li> <li>Activation automatique</li> </ul> |

## OPERATEURS D'ACTIVATION

|   |   |  |
|---|---|--|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 1.2em;">Opérateurs d'activations</span> </div>  |   | <p><b>Pour les conditions d'activation avec demande de confirmation ou automatique :</b></p> |
| <ul style="list-style-type: none"> <li><span style="color: blue;">1</span> Un seul état</li> <li><span style="color: blue;">📄</span> composition alternative</li> <li><span style="color: blue;">📄</span> composition normale</li> <li><span style="color: blue;">📄</span> composition exclusive</li> <li><span style="color: blue;">🕒</span> Un temps</li> </ul> | <ul style="list-style-type: none"> <li>Condition d'activation avec l'état d'un équipement comme déclencheur</li> <li>Composition alternative de plusieurs conditions d'activation</li> <li>Composition normale de plusieurs conditions d'activation</li> <li>Composition exclusive (ex : l'une ou l'autre, mais pas les deux)</li> <li>Condition d'activation par rapport au temps (heure, plage horraire)</li> </ul> |  |

## TYPE D'ACTIONS (SERVICES/MODES)

|  |   |
|--|---|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 1.2em;">Type d'actions</span> </div>   |   |
| <ul style="list-style-type: none"> <li><span style="color: green;">📄</span> Service élémentaire</li> <li><span style="color: green;">📄</span> Scénario (composition)</li> <li><span style="color: green;">📄</span> Service d'objectif</li> <li><span style="color: red;">📄</span> Une Interdiction</li> <li><span style="color: red;">📄</span> Interdiction (composition)</li> <li><span style="color: red;">📄</span> Interdiction d'objectif</li> </ul> | <ul style="list-style-type: none"> <li>Service élémentaire associé à une opération</li> <li>Composition de services : scénario comprenant des opérateurs</li> <li>Service d'objectif (dépasse le cadre du tutoriel)</li> <li>Mode avec une seule interdiction</li> <li>Mode avec plusieurs interdictions</li> <li>Mode d'objectif (dépasse le cadre du tutoriel)</li> </ul> |

## CONNEXIONS

|  |   |
|--|---|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 1.2em;">Connexions</span> </div>                                       |   |
| <ul style="list-style-type: none"> <li><span style="color: blue;">↗</span> activation-action</li> <li><span style="color: red;">↗</span> service alternatif</li> </ul> | <ul style="list-style-type: none"> <li>Association entre une condition d'activation et une action</li> <li>Définition d'un service alternatif (mauvais résultat d'exécution)</li> </ul> |

## OPERATEURS POUR LES SCENARIOS

|  |  |
|--|--|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <span style="font-size: 1.2em;">Opérateurs de composition</span> </div>  |  |
| <ul style="list-style-type: none"> <li><span style="color: blue;">○</span> Etape (service)</li> <li><span style="color: blue;">—</span> Transition</li> <li><span style="color: blue;">↗</span> transition-étape</li> <li><span style="color: blue;">→</span> étape-transition (réussite)</li> <li><span style="color: blue;">→</span> étape-transition (échec)</li> <li><span style="color: blue;">→</span> étape-transition (indiff.)</li> <li><span style="color: green;">—</span> Marqueur début</li> <li><span style="color: red;">—</span> Marqueur fin</li> </ul> | <ul style="list-style-type: none"> <li>Etape de composition : lien vers un service</li> <li>Transition entre deux étapes (permet de définir des temps d'attente entre activations de services)</li> <li>Lien entre une transition et une étape (transition vers étape)</li> <li>Lien entre une étape et une transition en considérant l'exécution du service correcte</li> <li>Lien entre une étape et une transition en considérant l'exécution du service incorrecte</li> <li>Lien entre une étape et une transition sans prendre en compte le résultat d'exécution</li> <li>Marqueur de début obligatoire pour tout scénario</li> <li>Marqueur de fin obligatoire pour tout scénario</li> </ul> |

FIGURE 6 – Objets graphiques présents dans la palette de la vue interaction

---

# IntelHome Designer version 1.3

Appartements tremplins du CMRRF de Kerpape

---

**Étude expérimentale pour l'évaluation de l'outil de CAO**

Willy Allègre (Lab-STICC / Université de Bretagne Sud)

---

# Introduction

## Présentation d'*IntelHome Designer*

Il s'agit de la première version d'*IntelHome Designer*, actuellement en cours de développement. Ce logiciel vise à rendre la conception de systèmes domotiques<sup>1</sup> (actuellement réservée à des techniciens/domoticiens) accessible à des personnes non expertes de ce domaine. Que ce soit un ergothérapeute (cadre institutionnel) ou bien un membre de la famille (cadre du maintien à domicile), cette personne semble être la mieux placée pour prendre en compte les besoins de vie exprimés par l'utilisateur en situation de handicap.

## Présentation du cas d'étude

Le cas d'étude reprend les appartements tremplins du CMRRF<sup>2</sup> de Kerpape, conçu pour évaluer l'aptitude au retour à domicile de personnes en situation de handicap. Entièrement équipés en domotique, ces appartements servent de cadre pour tester l'autonomie des patients en situation réelle. Pour ce cas d'étude, nous considérons deux utilisateurs qui occupent ces appartements<sup>3</sup> :

- L'*utilisateur 1* occupe l'*appartement 1* (studio de 35m<sup>2</sup>), équipé d'un bus domotique propriétaire
- L'*utilisateur 2* occupe l'*appartement 2* (T2 de 75m<sup>2</sup>), équipé d'un bus domotique ouvert

## Objectif du tutoriel

À ce stade, l'installation et l'intégration domotique des appartements (équipements, câblage, etc.) a déjà été réalisé par un domoticien. Il faut donc à présent adapter l'environnement de vie aux besoins des utilisateurs présentés dans la section précédente. Pour cela, différentes activités de modélisation sont proposées<sup>4</sup> :

1. *ENV* : la modélisation de l'environnement domotique et ses équipements (pour l'*appartement 1*)
2. *INT* : la modélisation des interactions utilisateur-système (pour l'*utilisateur 1* et l'*utilisateur 2*)

**Cette expérimentation consiste donc à se mettre dans la peau du concepteur non-expert, chargé de définir un environnement de vie adapté aux besoins de ces utilisateurs en utilisant le logiciel *IntelHome Designer* et en suivant ce tutoriel.**

L'expérimentation comprend 3 parties. La première partie vous permettra de découvrir le logiciel sur un exemple simple, avant de passer au cas d'étude réel de ce tutoriel. Pour chaque de ces parties, vous devrez suivre différentes étapes de la conception domotique : l'importation d'un fichier de configuration, la modélisation de l'environnement domotique, la modélisation des interactions entre l'utilisateur et son système et la génération automatique du fichier de configuration de l'application domotique. À la fin du tutoriel, un questionnaire sera distribué pour recueillir vos impressions sur l'utilisation du logiciel.

---

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. Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles de Kerpape : <http://www.kerpape.mutualite56.fr/>

3. Les appartements tremplins n'étant pas en fonction à l'heure actuelle, il s'agit d'un scénario fictif

4. Ce tutoriel propose de suivre et d'utiliser la méthode de modélisation et le langage graphique de description de systèmes domotiques proposés dans "*Model-Driven Flow for Home Automation System Design*", W. Allègre, T. Burger, P. Berruet.



# 1 Découverte du logiciel

L'exemple présenté dans cette section a pour but de se familiariser avec l'utilisation du logiciel. Il s'agit de définir l'environnement domotique présenté dans la figure 1 composé de 2 pièces et 4 équipements (2 lumières et une porte), puis de définir deux services. Le code de l'application domotique est par la suite automatiquement généré : l'application domotique est correctement configurée et prête à l'emploi.

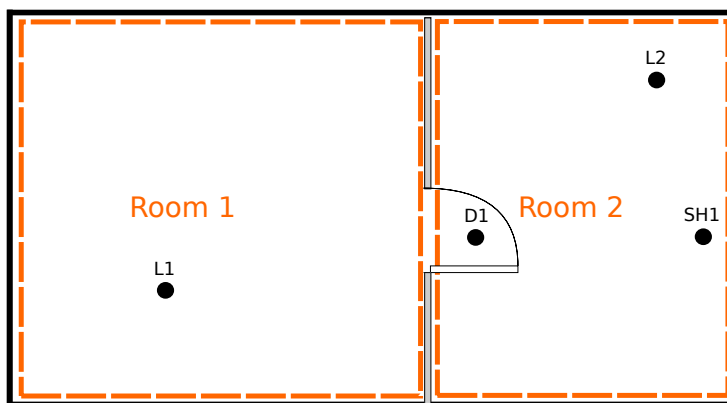


FIGURE 1 – Plan de l'environnement domotique à modéliser

**Question 1** : Prenez connaissance de l'organisation générale du logiciel présenté en annexe A.

**Question 2** : Prenez connaissance du tableau de bord IntelHome qui va permettre de vous guider dans les différentes étapes de la conception (annexe B). Ces étapes sont reprises dans la suite du tutoriel pour l'exemple illustratif et le seront pour le cas d'étude réel.

## 1.1 Étape 1 : Importation du fichier de configuration

Un fichier de configuration a été défini par les domoticiens lors de l'installation et l'intégration du bus domotique et des différents équipements. Il permet de lier des commandes (ex : infrarouge) aux équipements présents dans l'habitat.

**Question 3** : À partir du tableau de bord, sélectionnez le fichier de configuration "*exemple.conf*", puis cliquez sur la flèche *Importer* pour passer à l'étape suivante.

## 1.2 Étape 2 : Modélisation de l'environnement domotique

**Équipements et pièces** : Le premier travail du concepteur non-expert consiste à modéliser l'environnement domotique : on y définit les équipements (portes, lumières, etc.) et les pièces de l'habitat.

**Question 4** : À partir du tableau de bord, cliquez sur le lien *Édition* pour commencer la modélisation de l'environnement domotique. À partir de la figure 1, définissez et nommez les pièces de l'habitat (Rappel : la vue *Properties* permet de modifier les paramètres des éléments du canevas sélectionnés).

**Question 5** : À partir de la figure 1 et du tableau 1, définissez et nommez les équipements domotiques (lumières *L1*, *L2*, porte *D1*, volets *SH1*). Les portes étant vues comme des interfaces entre deux pièces,

définissez pour chaque porte deux liens vers des pièces (paramètre *Door link* dans la fenêtre *Properties*).

|               | Lumière                | Porte                  | Volets                  |
|---------------|------------------------|------------------------|-------------------------|
| <i>Room 1</i> | <i>L1</i> <sup>1</sup> | <i>D1</i> <sup>1</sup> |                         |
| <i>Room 2</i> | <i>L2</i> <sup>2</sup> | <i>D1</i> <sup>1</sup> | <i>SH1</i> <sup>1</sup> |

TABLE 1 – Équipements et pièces (protocole utilisé : *Legrand* = 1, *infrarouge* = 2)

**Bornes domotiques** Pour commander ces équipements, il faut maintenant intégrer des bornes domotiques. Il s’agit de cartes électroniques miniatures qui peuvent commander les équipements de différentes façons : soit directement en utilisant un protocole de communication sans fil (ex : infrarouge, wifi...), soit en passant par un bus domotique qui permet de centraliser la commande de tous les équipements (ex : portes, volets, lumières...)). Pour cet exemple, une borne domotique est reliée à un bus pour contrôler la lumière *L1*, la porte *D1* et les volets *SH1*. Une autre borne peut commander directement la lumière *L2* en infrarouge.

**Question 6** : Définissez le bus domotique “*HAB 1*” dans la pièce *Room 1* avec les paramètres suivants :

- nom : *HAB 1*
- protocole de communication : *LeGrand*
- liens vers les équipements *LeGrand* (portant l’indice <sup>1</sup> dans le tableau 1)

**Question 7** : Définissez une borne domotique dans la pièce *Room 1* pour contrôler ce bus domotique :

- nom : *Server 1*
- protocole de communication : *LeGrand*
- lien vers le bus domotique *HAB 1*

**Question 8** : Définissez une deuxième borne domotique dans la pièce 2 pour contrôler la lumière *L2* en infrarouge :

- nom : *Server 2*
- protocole de communication : *Infrarouge*
- lien direct vers la lumière *L2* pour le contrôle en infrarouge

À ce stade, tous les équipements sont connectés à une borne domotique : soit directement, soit par l’intermédiaire d’un bus. La lumière *L2* est directement connectée à la borne *Server 2* pour le contrôle en infrarouge. La lumière *L1*, la porte *D1* et les volets *SH1* sont connectés à la borne *Server 1* par l’intermédiaire du bus domotique *HAB 1*.

### 1.3 Étape 3 : Génération automatique du modèle d’interactions

Avant de passer à la modélisation des interactions entre l’utilisateur et son système domotique, il faut valider le modèle d’environnement défini dans l’étape précédente.

**Question 9** : À partir du tableau de bord (vue *DashBoard*), validez le modèle d’environnement. Une fenêtre s’affiche pour donner le résultat de cette validation. S’il y a des erreurs dans le modèle défini (ex : oubli du nom pour un équipement), une fenêtre s’affiche pour proposer des corrections à apporter. Pour passer à l’étape suivante, cliquez sur la flèche *Dériver*. (Tant que les erreurs ne sont pas toutes corrigées, il est impossible de passer à l’étape suivante.)

## 1.4 Étape 4 : Modélisation des interactions utilisateur-système

À ce stade, un modèle minimaliste d'interactions a été automatiquement généré à partir des équipements présents dans l'environnement domotique. Ce modèle comprend des services simples (ex : allumer lumière *L1*) activables manuellement. Il est possible d'enrichir ce modèle pour prendre en compte les besoins des utilisateurs. Par exemple, on peut y définir des services avec condition d'activation automatique, avec demande de confirmation de l'utilisateur, ou encore de définir des modes d'interdiction. Pour cet exemple illustratif, trois interactions différentes seront modélisées : deux services et un mode d'interdiction.

**Question 10** : Prenez connaissance des éléments graphiques pour définir des interactions (annexe E).

Dans un premier temps, l'utilisateur a besoin d'automatiser le fonctionnement de l'habitat. Pour cela, on peut définir des *scénarios* de vie (enchaînement de plusieurs services à partir d'une seule activation) avec des conditions d'activation automatique.

**Question 11** : À partir du tableau de bord (vue *DashBoard*), cliquez sur le lien *Édition* pour commencer la modélisation des interactions puis définissez le service suivant (cf exemples de l'annexe F) :

- nom : *Soir*
- condition d'activation : automatique lorsqu'il est 22h
- action : composition
  - allumer la lumière *L1*
  - descendre les volets *SH1*

Pour prévenir des éventuels pannes des actionneurs, il est possible de définir des scénarios avec prise en compte du résultat d'exécution.

**Question 12** : Définissez le service suivant (cf exemples de l'annexe F) :

- nom : *Soir2*
- condition d'activation : demande de confirmation lorsque les volets *SH11* sont descendus
- action : composition
  - allumer la lumière *L1*
  - si la lumière s'est correctement activée → fin du scénario
  - sinon allumer la lumière de la pièce 2 (*L2*)

De manière complémentaire aux services, il est possible de définir des interdictions sur l'environnement domotique. La définition de modes d'interdiction permet ainsi de restreindre momentanément l'accès aux services domotiques depuis l'interface de l'utilisateur.

**Question 13** : Définissez le mode d'interdiction suivant (cf exemples de l'annexe F) :

- nom : *AntiFermeture*
- condition d'activation : automatique entre 22h et 8h
- action : interdiction
  - interdire de fermer la porte *D1*

## 1.5 Étape 5 : Génération automatique du fichier de configuration

L'activité de modélisation est à présent terminée : l'environnement domotique et les interactions ont été définis. Il faut maintenant générer le fichier de configuration pour que l'application domotique *Danah*<sup>5</sup>

---

5. *Danah* est une application domotique développée au laboratoire *Lab-STICC* qui permet de contrôler différents équipements en infrarouge ou via un bus domotique (ex : *KNX* et/ou *LeGrand*)

prennent en compte le travail de modélisation effectué.

**Question 14** : À partir du tableau de bord, validez le modèle d’interactions, puis cliquez sur la flèche *Combiner*. (Tant que les erreurs ne sont pas toutes corrigées, il est impossible de passer à l’étape suivante.)

Avant de générer le modèle spécifique à la plateforme Danah, il faut lier les équipements et leurs opérations définies dans le modèle d’environnement avec des labels et commandes physiques.

**Question 15** : À partir du tableau de correspondance 2 et 3, définissez pour chaque équipement et pour chaque opération un label et une commande physique.

| Équipement | Label | Équipement | Label |
|------------|-------|------------|-------|
| L1         | GEWA1 | D1         | GEWA2 |
| L2         | GEWA1 | SH1        | GEWA2 |

TABLE 2 – Correspondances entre équipement-label

| Opération           | Commande    | Opération            | Commande |
|---------------------|-------------|----------------------|----------|
| Allumer_Lumière_L1  | GEWA1.key_1 | Ouvrir_D1            | GEWA2.A  |
| Éteindre_Lumière_L1 | GEWA1.key_2 | Fermer_D1            | GEWA2.B  |
| Allumer_Lumière_L2  | GEWA1.key_3 | Monter_Volets_SH1    | GEWA2.C  |
| Éteindre_Lumière_L2 | GEWA1.key_4 | Descendre_Volets_SH1 | GEWA2.D  |

TABLE 3 – Correspondance entre opération-commande

**Question 16** : Une fois les liens définis, validez le modèle *Danah* puis cliquez sur la flèche *Dériver* pour générer le fichier de configuration du système domotique. Pour avoir un aperçu du fichier généré à partir des modèles que vous avez défini, vous pouvez *éditer* le fichier.

## A Présentation du logiciel

Le logiciel *IntelHome Designer* est composé de différentes vues (cf **figure 3**) :

1. Le Canevas permet de manipuler les éléments graphiques d'un modèle (environnement ou interactions)
2. La Palette permet d'importer des éléments graphiques pour les mettre sur le canevas
3. La vue Properties permet de définir, pour chaque objet sélectionné, ses propriétés
4. La vue DashBoard permet de guider l'utilisateur dans les étapes de la conception (cf annexe B)

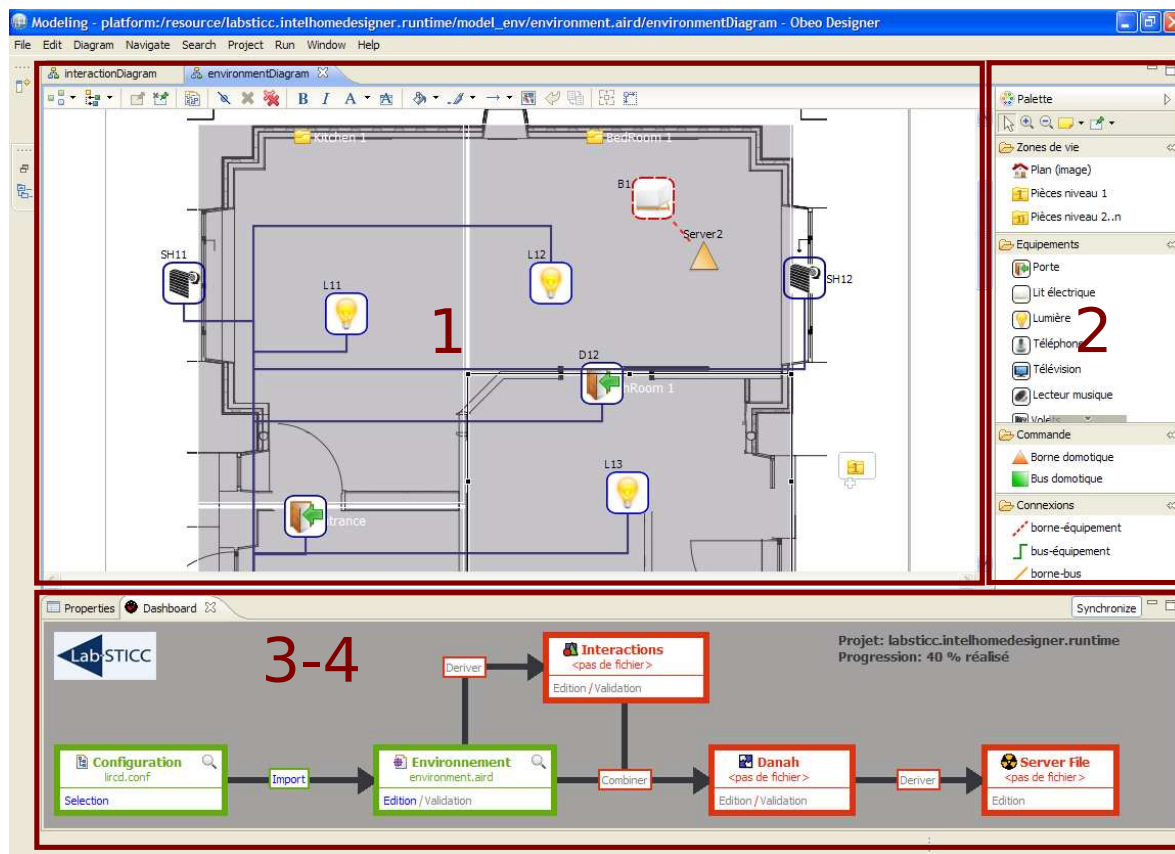


FIGURE 2 – Organisation générale du logiciel IntelHome Designer

Au dessus du canevas, des filtres et des calques peuvent être ajoutés/retirés pour cacher certains éléments graphiques du canevas ou de la palette :

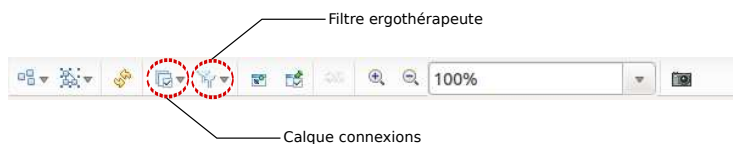


FIGURE 3 – Menu pour la modélisation de l'environnement (filtre / calque)

## B Présentation du tableau de bord

Le tableau de bord de la vue **DashBoard** reprend les différentes étapes de la conception domotique :

1. Importation du fichier de configuration
2. Modélisation de l'environnement domotique (avec la palette)
3. Génération automatique du modèle d'interactions
4. Modélisation des interactions utilisateur-système (avec la palette)
5. Génération automatique du fichier de configuration

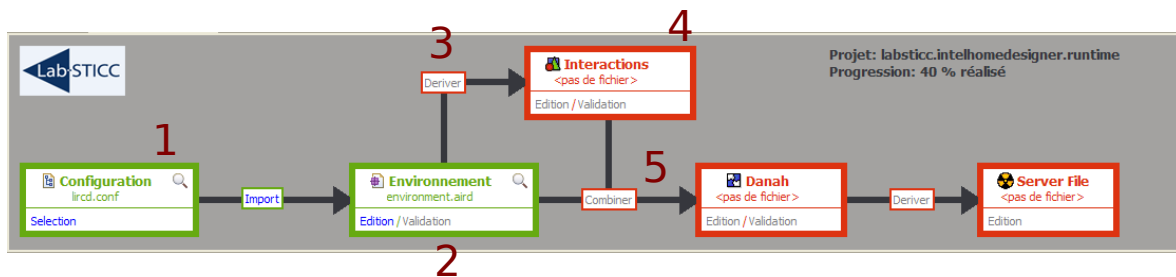
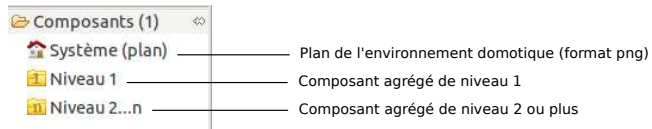


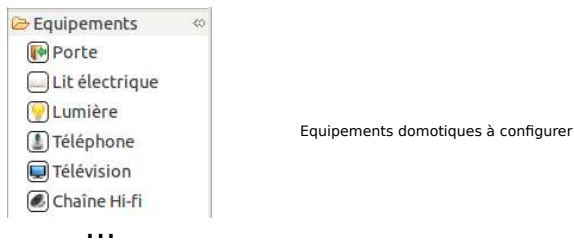
FIGURE 4 – Tableau de bord visuel pour guider l'utilisateur dans les étapes de la conception domotique

## C Éléments graphiques pour la modélisation de l'environnement

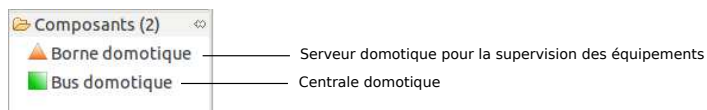
### COMPOSANTS AGREGES



### COMPOSANTS ELEMENTAIRES



### ORGANES DE COMMANDE



### CONNEXIONS (ORGANES DE COMMANDE)

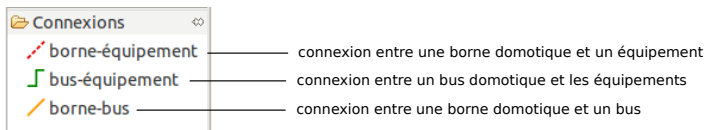


FIGURE 5 – Éléments graphiques présents dans la palette (modèle d'environnement)

# D Éléments graphiques pour la modélisation d'interactions

## TYPE D'INTERACTIONS

|  |  |
|--|--|
|  | <b>Interaction = Condition d'activation + Action</b> |
|  | Profil lié à un utilisateur                          |
|  | Service (permet d'exprimer la formule "je veux")     |
|  | Mode (permet d'exprimer la formule "j'interdis")     |

## TYPE DE CONDITIONS D'ACTIVATION

|  |   |
|--|---|
|  |   |
|  | Activation sous demande utilisateur                 |
|  | Activation automatique avec demande de confirmation |
|  | Activation automatique                              |

## OPERATEURS D'ACTIVATION

|  |   |
|--|---|
|  | <b>Pour les conditions d'activation avec demande de confirmation ou automatique :</b> |
|  | Condition d'activation avec l'état d'un équipement comme déclencheur                  |
|  | Composition alternative de plusieurs conditions d'activation                          |
|  | Composition normale de plusieurs conditions d'activation                              |
|  | Composition exclusive (ex : l'une ou l'autre, mais pas les deux)                      |
|  | Condition d'activation par rapport au temps (heure, plage horraire)                   |

## TYPE D'ACTIONS (SERVICES/MODES)

|  |  |
|--|--|
|  |  |
|  | Service élémentaire associé à une opération                  |
|  | Composition de services : scénario comprenant des opérateurs |
|  | Service d'objectif (dépasse le cadre du tutoriel)            |
|  | Mode avec une seule interdiction                             |
|  | Mode avec plusieurs interdictions                            |
|  | Mode d'objectif (dépasse le cadre du tutoriel)               |

## CONNEXIONS

|  |   |
|--|---|
|  |   |
|  | Association entre une condition d'activation et une action        |
|  | Définition d'un service alternatif (mauvais résultat d'exécution) |

## OPERATEURS POUR LES SCENARIOS

|  |  |
|--|--|
|  |  |
|  | Etape de composition : lien vers un service  |
|  | Transition entre deux étapes (permet de définir des temps d'attente entre activations de services) |
|  | Lien entre une transition et une étape (transition vers étape)                                     |
|  | Lien entre une étape et une transition en considérant l'exécution du service correcte              |
|  | Lien entre une étape et une transition en considérant l'exécution du service incorrecte            |
|  | Lien entre une étape et une transition sans prendre en compte le résultat d'exécution              |
|  | Marqueur de début obligatoire pour tout scénario   |
|  | Marqueur de fin obligatoire pour tout scénario   |

FIGURE 6 – Éléments graphiques présents dans la palette (modèle d'interactions)