

# Table des matières

<b>Introduction</b>	<b>6</b>
<b>1 Les technologies de la domotique : Etat de l'art</b>	<b>8</b>
1.1 Introduction	8
1.2 Qu'est ce que la domotique	8
1.2.1 Définition	8
1.2.2 Origines	9
1.2.3 Pourquoi la domotique	9
1.2.4 Exemples de scénarios	10
1.3 Matériels et technologies de support de transport	11
1.3.1 Les technologies filaire	12
1.3.2 Les technologies sans fil	14
1.4 Les technologies de découverte de services	16
1.4.1 JINI	16
1.4.2 SLP : Service Location Protocol	17
1.4.3 UPNP : Universal Plug and Play	17
1.5 Problématique	18
1.6 Conclusion	18
<b>2 Spécification des besoins</b>	<b>20</b>
2.1 Introduction	20
2.2 Objectifs	20
2.3 Etude de l'existant	21
2.4 La spécification des exigences	22
2.4.1 Hétérogénéité	22
2.4.2 Nommage et adressage	23
2.4.3 La mobilité	23
2.4.4 La fiabilité	24
2.4.5 Le mode de communication	24
2.4.6 Absence d'administrateur	24
2.5 Les besoins fonctionnels	25
2.6 Structure et solution retenue	25
2.7 Conclusion	26

<b>3</b>	<b>Le protocole SIP</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Structure générale du protocole SIP . . . . .	27
3.3	Architecture du protocole SIP . . . . .	28
3.4	Les fonctions utilisées par SIP . . . . .	28
3.5	Les messages SIP . . . . .	30
3.5.1	Les requêtes de base SIP . . . . .	30
3.5.2	Les autres requêtes SIP . . . . .	31
3.5.3	Les réponses SIP . . . . .	31
3.5.4	Format des messages SIP . . . . .	31
3.6	Le protocole SIP et la domotique . . . . .	32
3.6.1	Comparaison SIP / HTTP . . . . .	34
3.6.2	Comparaison SIP / SNMP . . . . .	34
3.6.3	Comparaison SIP / SMTP . . . . .	34
3.7	Conclusion . . . . .	35
<b>4</b>	<b>Conception de l'application</b>	<b>36</b>
4.1	Introduction . . . . .	36
4.2	Méthodes de conception . . . . .	36
4.2.1	Méthode fonctionnelle . . . . .	36
4.2.2	Méthode orientée objets . . . . .	37
4.3	Conception de la base de données . . . . .	37
4.3.1	Le niveau conceptuel : Analyse . . . . .	37
4.3.2	Modèle conceptuel de donnée (MCD) . . . . .	40
4.3.3	Modèle Logique de données . . . . .	41
4.4	Conception de l'application . . . . .	42
4.4.1	Méthodologie adoptée . . . . .	42
4.4.2	Représentation des diagrammes des cas d'utilisations . . . . .	44
4.4.3	Représentation des diagrammes de séquences . . . . .	45
4.5	Conclusion . . . . .	48
<b>5</b>	<b>Réalisation</b>	<b>49</b>
5.1	Introduction . . . . .	49
5.2	Environnement matériel et logiciel . . . . .	49
5.2.1	Environnement matériel . . . . .	50
5.2.2	Environnement logiciel . . . . .	52
5.3	Implémentation . . . . .	54
5.3.1	Développement de la passerelle SIP . . . . .	54
5.3.2	Développement des modules ZigBee . . . . .	54
5.4	Validation de l'application . . . . .	54
5.5	Scénarios d'utilisations de la plateforme domotique . . . . .	55
5.5.1	Découverte des listes des capteurs . . . . .	55
5.5.2	Déterminer la valeur d'un capteur . . . . .	57
5.5.3	Découverte des dispositifs X10 . . . . .	58
5.5.4	Commande des dispositifs X10 . . . . .	59
5.6	Conclusion . . . . .	59

**Conclusion générale**

**61**

# Table des figures

1.1	Applications domotiques . . . . .	11
1.2	Un système X.10 . . . . .	12
1.3	Une installation EIB . . . . .	14
1.4	Problème d'accès dans un réseau domotique . . . . .	18
2.1	Plateforme HomeSIP . . . . .	26
3.1	Positionnement du protocole SIP parmi les protocoles Internet . . . . .	28
3.2	Infrastructure SIP . . . . .	30
3.3	Échange des messages SIP pour une communication VoIP . . . . .	32
3.4	Utilisation de SIP dans un environnement domotique . . . . .	33
4.1	Roues codeuses pour identification d'un dispositif X10 . . . . .	38
4.2	Cas d'utilisation général de notre système domotique . . . . .	44
4.3	Cas d'utilisation «Piloter le réseau domotique» . . . . .	45
4.4	Diagramme de séquence «Enregistrement de l'utilisateur distant» . . . . .	46
4.5	Diagramme de séquence «Découverte des dispositifs domotiques» . . . . .	46
4.6	Diagramme de séquence «Notification sur les états des dispositifs» . . . . .	47
4.7	Diagramme de séquence «Déterminer l'état courante d'un dispositif» . . . . .	47
4.8	Diagramme de séquence «Positionner la valeur d'un dispositif» . . . . .	48
4.9	Diagramme de séquence «Activation/désactivation d'un dispositif» . . . . .	48
5.1	Matérialisation de l'architecture HomeSIP . . . . .	50
5.2	Microcontrôleur ARM09 . . . . .	50
5.3	Capteur iButton DS1920 avec adaptateur DS9097U . . . . .	51
5.4	Module XBee . . . . .	51
5.5	Module CM11 . . . . .	52
5.6	Module LM11 . . . . .	52
5.7	strategie de test adoptée . . . . .	55
5.8	Structure SOAP de la commande «getListeCapteur» . . . . .	56
5.9	Trace de la commande «getListeCapteur» . . . . .	56
5.10	Réponse SIP à la commande «getListeCapteur» . . . . .	57
5.11	Trace de la commande «getValeurCapteur» . . . . .	57
5.12	Réponse SIP à la commande «getValeurCapteur» . . . . .	58

5.13	Trace de la commande «getListeDispositifX10» . . . . .	58
5.14	Réponse SIP à la commande «getListeDispositifX10» . . . . .	59
5.15	Structure SOAP d'une commande X10 . . . . .	59

# Introduction

La rapidité des progrès technologiques dans le secteur des communications multimédia, en particulier la convergence de la télécommunication et des réseaux informatiques, a entraîné l'apparition d'une nouvelle classe d'applications parfois appelée « everywhere ». Ce néologisme est formé de 2 mots : « everywhere » et (hard/soft)ware, il englobe les termes informatique ambiante et informatique omniprésente (*ubiquitous computing*). Cela signifie étendre l'informatique, qui est pour l'instant limitée aux ordinateurs, aux objets qui nous entourent et qui n'ont à priori aucune vocation informatique, afin de faciliter les tâches usuelles de la vie de tous les jours.

Cette nouvelle classe d'applications consiste à rassembler, coordonner, transmettre, traiter et réagir aux informations émanant des entités de communication (ou des objets) fixes ou mobiles. Ces activités sont présentes dans divers domaines comprenant la médecine, l'automatique, la sûreté militaire et la domotique.

Ce nouveau type d'application a besoin d'une nouvelle plateforme pouvant gérer la coordination entre les différents périphériques (capteurs, caméras,...), appelé infrastructure de communication. La domotique nécessite des composants contrôlant la coordination entre les divers dispositifs câblés (sondes, appareils photos,...) et qui prennent en considération l'hétérogénéité des entités matérielles et des fonctionnalités des réseaux utilisés.

C'est dans ce cadre que nous avons réalisé notre projet au sein de l'équipe PHOENIX à l'INRIA Futurs de Bordeaux. Ce projet consiste à élaborer une plateforme domotique basée sur le protocole SIP. Initié début 2006, le projet HomeSIP en est à ses débuts ; il est en collaboration entre 2 groupes de travail des laboratoires de l'université Bordeaux I :

- Groupe Cosynux du laboratoire d'électronique IMS.
- Groupe PHOENIX du laboratoire d'informatique LABRI – INRIA.

Afin de présenter ce mémoire, nous avons choisi de le découper en cinq chapitres. Dans le premier chapitre, état de l'art, nous présentons les aspects (de point de vue technologie de conception et de réalisation) qui sont liées à notre projet. C'est-à-dire nous faisons une étude sur les produits similaires à notre application afin de tirer les apports de notre projet.

Dans le deuxième chapitre, analyse des besoins et spécification, nous analy-

sons les besoins fonctionnels de notre application en présentons la spécification des besoins et des exigences du produit.

Dans le troisième chapitre, le protocole SIP, nous présentons l'architecture du protocole SIP en indiquant les caractéristiques permettant son utilisation dans un environnement domotique.

Le quatrième chapitre, la conception, qui nous semble le plus important, sert à faire une étude détaillée sur les différents cas d'utilisations possibles afin de concevoir une application réalisable, fiable et efficace.

Le cinquième chapitre, la réalisation, est une introduction à l'application proprement dite, et ce en présentant l'environnement matériel et logiciel dans lesquels le projet a été réalisé, ainsi que les différents choix techniques et les stratégies de tests adoptés pour valider l'application.

Notre rapport sera achevé par la conclusion générale qui va résumer nos objectifs estimés et les limites de notre réalisation tout en évoquant les problèmes que nous avons rencontrés et les améliorations envisageables.

# Chapitre 1

## Les technologies de la domotique : Etat de l'art

### 1.1 Introduction

Les progrès technologiques de l'informatique, des télécommunications et de l'électronique ont permis le développement des systèmes de transmission, des commandes à distance et favorisé l'éclosion d'une offre abondante de nouveaux services pour les occupants des logements. Ces services, regroupés sous le terme "domotique", concernent principalement le confort (commande à distance d'appareils ou équipements,...), la sécurité (protection contre les intrusions, détection d'incendie, téléassistance aux personnes âgées ou en situation d'handicap), l'économie d'énergie (programmation, gestion du chauffage...). Le fonctionnement de ces services est fondé sur les réseaux de communication interne des habitations et sur leurs liaisons avec l'extérieur.

Dans ce chapitre, nous allons traiter ce nouveau domaine. Dans la partie une, nous détaillons la définition de la domotique ainsi que ses avantages. Ensuite, nous présentons les différentes technologies utilisées ce qui nous permettra de déceler les lacunes de la discipline et de présenter la solution qui nous paraîtra la plus adéquate.

### 1.2 Qu'est ce que la domotique

#### 1.2.1 Définition

La domotique rassemble les technologies de l'informatique, de l'électronique et des télécommunications utilisées dans les domiciles [1]. Un réseau domotique consiste en l'interconnexion d'équipements domestiques hétérogènes (ordinateurs, téléviseurs, assistants numériques personnels (PDA), matériel Hi-Fi, appareils électroménagers,...) dans le but de proposer des services à valeur ajoutée aux utilisateurs. Cette interconnexion permet par exemple d'accéder aux

contenus ou services multimédias présents sur le réseau depuis tout équipement en faisant partie. L'accès peut se faire soit depuis l'intérieur du domicile (sur des média filaires ou sans fil), soit depuis l'extérieur via l'Internet. La synchronisation automatique entre les équipements domotiques fixes et les réseaux personnels (PAN, Personal Area Network ) de chacun des habitants est un autre exemple de service augmenté offert par les réseaux domestiques.

### 1.2.2 Origines

Les réseaux domotiques trouvent leur origine dans le projet *Ubiquitous Computing* (informatique omniprésente) mené par *Marc Weizer* au *Xerox Parc* à la fin des années 80. Ce projet s'appuie sur le constat suivant : pour devenir omniprésente et véritablement utile, l'informatique doit devenir transparente [2] : les ordinateurs doivent être enfouis dans les objets du quotidien, communiquer, se configurer et fournir des services sans que l'utilisateur s'en aperçoive. Les réseaux domotiques sont une forme particulière d'*Ubiquitous Computing* se focalisant sur les lieux de résidence et leurs habitants. Plusieurs visions coexistent actuellement. Elles sont basées soit sur des technologies informatiques, soit sur des technologies issues de l'électronique grand public.

### 1.2.3 Pourquoi la domotique

Les motivations de l'utilisateur pour la domotique sont multiples et nous pouvons plus particulièrement considérer les points suivants : communication, surveillance, sécurité, gestion de l'énergie, confort et commodité.

#### La communication

Des nouveaux services comme les films ou l'information à la demande, le téléachat et la banque à domicile devraient être offerts aux utilisateurs dans un avenir proche. Toutes ces activités seront principalement organisées autour d'un téléviseur évolué, d'un téléphone mobile ou d'un PC.

#### La sécurité

En cas de menace pour la sécurité de la maison, tout composant domotique est capable d'émettre un message sur l'installation qui sera repris et traité par un module spécialisé pour la surveillance. Ce module peut alors déclencher n'importe quel composant présent dans l'installation afin de simuler une présence (lumière qui s'allume ou musique) ou bien renforcer la sécurité (verrouillage de toutes les serrures, déclenchement des alarmes). Ces actions peuvent se faire selon un choix particulier, selon une durée ou un nombre de détections ou bien directement par téléphone ou par un ordinateur à distance.

## La gestion de l'énergie

L'économie d'énergie justifie à elle-même l'investissement dans la gestion automatisée de l'habitat. Les deux éléments sur lesquels la domotique va influencer la dépense d'énergie sont :

- La distribution de chaleur dans les pièces.
- La suppression de la consommation électrique inutile.

Un système de régulation domestique est automatique et capable de faire face à tout évènement lié au chauffage. Le réglage des températures est simple et visuel, chaque pièce peut bénéficier d'un réglage qui lui est propre (absence/présence, jour/nuit) et une commande à distance par téléphone est possible afin de mettre la maison sur «confort» ou bien «économie» lors de l'absence des habitants. De plus, le gaspillage d'énergie peut être limité avec des produits domestiques afin de ne pas avoir de lampe oubliée à la cave pendant plusieurs jours, un éclairage surdimensionné ou une lampe allumée en plein jour.

## Le confort

L'accroissement du niveau de confort des habitations a été le premier objectif de la domotique. Les fonctions de commande à distance simples et qui agissent sur différents types d'appareils sont maintenant banalisées. En effet, il est possible d'activer à distance des fonctions pour recréer une ambiance ou un état prédéfinis dans la maison. Il est donc facile d'imaginer un nombre illimité des fonctions qui pourraient faciliter le confort quotidien dans la maison (par exemple la cafetière s'allume et les volets s'ouvrent à 7h tous les matins).

### 1.2.4 Exemples de scénarios

- **Le matin** : Ouverture des stores, enclenchement de la machine à café, chauffage mode "jour".
- **Départ au travail** : Mise en mode « économique » du chauffage dans les pièces inoccupées, contrôle des fenêtres ouvertes, extinction de toutes les lumières, enclenchement de la sécurité.
- **La soirée** : Diminution des lumières pour le Home Cinéma, contrôle des portes ouvertes (entrée, garage), fermeture des stores.
- **La nuit** : Extinction de toutes les lumières des pièces non utilisées, enclenchement de la sécurité, mise en mode « nuit » du chauffage selon l'usage individuel des pièces.

Tous ces scénarios peuvent être manipulés par un simple bouton, un PDA, un PC ou une télécommande. La figure 1.1 présente les applications les plus courantes comme :

- Sécurité des biens et des personnes ;
- Gestion d'ambiance (sonore ou lumineuse) ;
- Régulation de chauffage ou de climatisation.

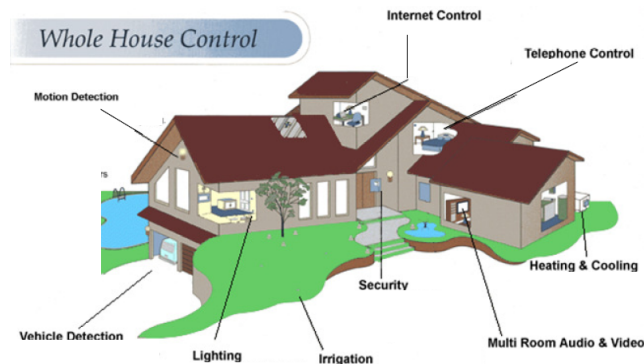


FIG. 1.1 – Applications domotiques

La domotique met en oeuvre des dispositifs que nous pouvons appeler des «appliance» qui sont une solution conjuguant hard et soft pour fournir une solution immédiatement opérationnelle. En général, il s'agit d'un ordinateur sommaire, tournant le plus souvent sous Linux, chargé d'une application directement utilisable. On trouve sous forme d'appliance, des serveurs e-Mail ou des boîtiers de sécurité réseau ou Internet [3].

### 1.3 Matériels et technologies de support de transport

Dans cette section, nous détaillons les différentes technologies appliquées à la domotique : les technologies de support de transport et les technologies de découverte des services.

Une première approche envisage les réseaux domotiques comme une évolution des LAN autour d'un élément fédérateur : l'ordinateur domestique. C'est l'approche des solutions telles que Universal Plug'n Play (UpnP). Elles permettent de gérer automatiquement l'installation et la configuration de nouveaux appareils fonctionnant sur IP et celle des services associés (partage de fichiers, d'imprimantes, de connexions à l'Internet) de manière transparente pour l'utilisateur. Une autre approche est plutôt héritée de l'électronique grand public : Les dispositifs sont tous connectés entre eux par un bus numérique au travers duquel ils échangent contenus, accès et traitements. Via ce bus, des équipements s'interconnectent facilement et adaptent conjointement leur configuration. La technologie HAVI (Home Audio-Video Interoperability) par exemple permet ce type d'interactions. Les deux approches ne s'excluent pas mutuellement, et certains efforts de recherche s'orientent aujourd'hui vers des approches hybrides. C'est le but de notre projet qui permettra l'interopérabilité entre des technologies domotiques différentes à travers l'utilisation d'une infrastructure SIP.

Les réseaux domotiques concernent les infrastructures de communication proposées traditionnellement pour le contrôle d'équipements d'habitation. Ils concernent aussi bien des technologies sans fils, radio ou infrarouge, ou encore des technologies filaires.

### 1.3.1 Les technologies filaire

#### X10

X10 est une technologie qui utilise le CPL (Courant Porteur en Ligne). Elle permet aux produits compatibles de dialoguer ensemble via les fils électriques du secteur de l'habitation. L'avantage est qu'il n'est pas nécessaire d'installer de nouveaux câbles de communication, on réutilise ceux déjà en place (les fils par lesquels arrive le courant électrique), et ceci sans dégradation. Le signal de communication est superposé à la tension 110 Volts ou 220 Volts. Le protocole X10 régit la communication entre les commandes (télécommandes, boîtiers de contrôle ou ordinateurs) et les différents équipements de la maison. Pour cela, on place entre les appareils à commander et les prises de courant des modules de communications [4].

La norme X10 utilise des transmetteurs spécifiques qui émettent un signal codé à travers le réseau électrique de la maison. Les récepteurs X10 branchés entre les appareils et les prises de courant détectent ce signal et agissent en fonction du message reçu. Le contrôle des messages est effectué grâce à un tableau de commande, une télécommande ou encore par un ordinateur, relié à un transmetteur qui relaiera les ordres aux différents modules branchés sur le réseau électrique. La figure 1.2 représente les différents équipements dans une installation X10.



FIG. 1.2 – Un système X.10

La technologie X10 possède les avantages suivants :

- Contrôle automatique des lumières extérieures ou des lampes de sécurité.
- Simulation de présence dans la maison en cas d'absence.

- Commande de toutes les lumières intérieures et extérieures avec les télécommandes RF.
- Programmation des scénarios pour un système X10 avec lequel il est possible d'exécuter plusieurs commandes simultanément comme par exemple "Allumer toutes les lampes" ou " Scénario matin " suivant un mode de vie personnelle.
- Installation d'interrupteurs grâce aux interrupteurs sans fil dans n'importe quel endroit où il n'y a pas de câbles.
- Adaptation des maisons pour des personnes âgées ou des personnes handicapées d'une façon simple et flexible. Grâce aux télécommandes sans fil ils peuvent commander tout en poussant sur un seul bouton.

La domotique se trouve au carrefour de l'informatique, de l'électronique et du contrôle des maisons. Dans ce cadre, le protocole X10 se positionne comme une solution simple, économique et efficace. Cette technologie commence à se développer et, de plus en plus, des solutions intéressantes sont proposées dans le commerce. Le X10 se positionne comme une technologie puissante et bon marché avec de grandes perspectives d'évolution de par sa facilité de mise en oeuvre. Nous détaillons la technologie X10 dans l'annexe A.

## **HAVi**

Home Audio Video interoperability est le nom de l'organisation créée par *Grundig, Hitachi, Matsushita/Panasonic, Philips, Sharp, Sony, Thomson multimedia* et *Toshiba* pour développer des spécifications visant à faciliter la communication entre équipements grand public audiovisuels et multimédias dans la maison. Publiée en 2000, la spécification HAVI 1.0 définit un ensemble de modules logiciels (API et middleware) qui automatisent l'échange des messages entre équipements et la mise en commun de leurs ressources par le biais du bus série. Toute application tournant sur un produit HAVI est alors capable de détecter et d'utiliser une fonction offerte par un autre matériel connecté au réseau, et ce, quelle que soit leur marques respectives[5].

## **CEBus**

CEBus [6] (Consumer Electronics Bus) est un standard de communication développé par l'EIA (Electronics Industry Association) et le CEMA (Consumer Electronics Manufacturers Association) et approuvé en 1992. Ce standard est ouvert et par conséquent tout le monde peut l'utiliser. La norme ne s'applique pas simplement à la transmission par courant porteur mais également à la transmission par câble coaxial, RF et infrarouge. L'inconvénient de CEBus est qu'il y'a relativement peu de produits disponibles et le coût de ces produits est élevé.

## **EIB**

Le standard EIB [7] (European Industrial Bus) est normalisé ISO (International Standardisation Organisation). C'est un système ouvert ; il regroupe plus

de 23 organisations nationales. Il couvre tous les besoins concernant l'habitat et le bâtiment en matière de confort, d'économie d'énergie et aussi de sécurité. Contrairement à une installation traditionnelle, Dans une installation domotique EIB, seuls les éléments qui ont besoin d'énergie sont reliés au 220V. Tous les interrupteurs et autres capteurs présents ne sont reliés que par un seul câble EIB 29V.

D'après la figure 1.3, un système EIB se compose de deux éléments, les actionneurs (exécuteurs d'ordres, citons par exemple : les lampes, stores, vannes, moteurs, prises de courant) et les capteurs (transmetteurs d'ordres, citons par exemple : les interrupteurs, les écrans de commandes, les sondes,...), sans oublier les systèmes «sans fils».

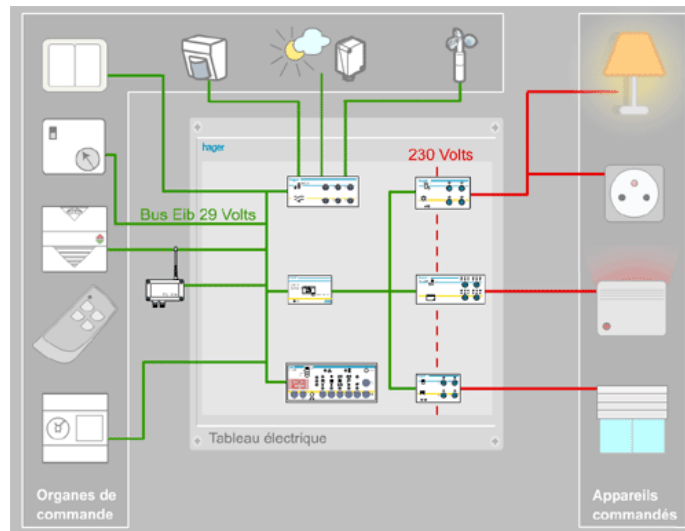


FIG. 1.3 – Une installation EIB

### 1.3.2 Les technologies sans fil

#### HomeRF

HomeRF est une spécification de réseau sans fil (Shared Wireless Access Protocol-SWAP) permettant à des périphériques domestiques échanger des données entre eux. Elle a été mise au point par le *HomeRF Working Group*, un groupe de sociétés actives dans le réseau sans fil incluant *Siemens*, *Motorola* et plus de cent autres sociétés. Le groupe a été dissout en janvier 2003 lorsque la norme Wi-Fi IEEE 802.11 est devenue disponible pour des usages domestiques et que Microsoft a choisi Bluetooth, concurrent direct de HomeRF, dans ses systèmes Windows, ce qui provoqua le déclin de cette spécification[8].

## Bluetooth

Bluetooth, inventé par *Ericsson* et adapté par *Toshiba, IBM, Intel* et *Nokia*, est un système de transmission et de réception simultanée ou bien séparée aussi bien des données que de voix. La technologie Bluetooth a pour objectif principal de substituer le câblage entre équipements électroniques, informatiques et téléphoniques par un lien radio universel (bande ISM à 2.4 GHz.) courte portée à faible consommation d'énergie. Les équipements en question peuvent être des terminaux téléphoniques, des matériaux électroniques, des PC et leurs périphériques.

Bluetooth, qui s'adapte à l'environnement domestique, permet ainsi de créer de petits réseaux d'équipements (téléphone, ordinateurs, PDA,...) sans avoir les contraintes d'infrastructures fixes des réseaux habituellement mis en œuvre.

Les propriétés Bluetooth permettent de gérer un ensemble considérable de dispositifs dans le domicile tout en offrant la possibilité de plusieurs connexions simultanées, ce qui permet par exemple la gestion des alarmes et appels d'urgence pendant l'exécution d'une commande. Malheureusement, le grand défaut de cette technologie est sa trop grande consommation d'énergie. Elle ne peut donc pas être utilisée par des capteurs qui sont alimentés par une batterie et qui, idéalement, devraient fonctionner durant plusieurs années.

## ZigBee

Zigbee est une technologie radio basée sur le standard IEEE 802.15.4 et destinée en premier lieu au contrôle, à la surveillance et à la gestion des commandes à distance. Des applications peuvent par exemple être trouvées dans le domaine de la domotique tel que :

- Activation d'un éclairage à distance.
- Surveillance d'un bâtiment.
- Relevé d'informations transmises par des capteurs.

Cette technologie convient particulièrement au marché des bâtiments commerciaux. Le standard ZigBee offre d'une part des caractéristiques qui répondent aux besoins des réseaux domotiques et d'autre part il garantit des débits de données moindres mais consomme également nettement moins d'énergie que Bluetooth. En effet, un petit débit de données n'est pas handicapant pour un réseau de capteurs où les débits de transmission ne sont pas soutenues et conséquentes. En mettant l'accent sur la fiabilité, le faible coût, la longue durée de vie des piles et la facilité du déploiement, ZigBee prépare le terrain pour permettre à des capteurs intelligents d'offrir une efficacité accrue de l'énergie et un meilleur contrôle des systèmes d'éclairage, de chauffage, de climatisation et de sécurité dans les bureaux et dans leurs environs.

Malgré tout, la tendance actuelle des constructeurs est d'employer des technologies propriétaires qui ont pour avantage d'être spécifiquement optimisées pour une utilisation précise mais qui ont comme gros inconvénient de ne pas être compatibles entre elles.

De nouvelles technologies vont influencer considérablement l'avenir des réseaux

domotiques. UWB (Ultra Wide Band) en est un très bon exemple. Cette technique de transmission permettra d'atteindre des niveaux de consommation extrêmement bas grâce à sa simplicité au niveau matériel. De plus, l'atténuation du signal engendré par des obstacles est moindre qu'avec les systèmes radio à bande étroite conventionnels.

Il existe une certification ZigBee qui garantit l'interopérabilité des solutions matérielles et logicielles venant de multiples fournisseurs. C'est la ZigBee Alliance[9] qui assure le développement et la promotion de la technologie ZigBee. Il constitue un groupe de certification pour la norme IEEE 802.15.4. L'Alliance ZigBee est une association d'entreprises qui collaborent dans la conception des produits de supervision et de contrôle réseau sans fil, fiables, économiques et à faible consommation, fondés sur un standard ouvert mondial. L'Alliance ZigBee se compose des fournisseurs de technologies et des fabricants d'équipements dans le monde entier.

## 1.4 Les technologies de découverte de services

Les protocoles de découverte de services étudiés ici ont pour objectif de permettre à un équipement ou un logiciel de trouver les services du réseau domotique d'une manière automatique sans que la configuration manuelle soit nécessaire. Par exemple, les protocoles de découverte des services (Jini[10], SLP[11], UPnP[12],...) fournissent les mécanismes automatiques pour localiser les services offerts dans un réseau. Ces protocoles permettent aux serveurs de publier leurs services, et aux clients de découvrir ces services.

### 1.4.1 JINI

Jini est une technologie développée et distribuée par Sun Microsystems. Elle offre une infrastructure logicielle permettant à des objets Java (services) de se découvrir et de s'utiliser de façon spontanée.

Un réseau Jini est constitué des entités suivantes :

- **Proxy** : c'est un objet Java sérialisable qui assure la connexion aux services est les invocations des méthodes (peut être un stub RMI).
- **Service lookup (LUS)** : les services enregistrent leurs Proxy auprès des LUS.
- **Service** : il prépare un Proxy d'accès au service, recherche un LUS et enregistre le service auprès du LUS.
- **Client** : interroge tous les LUS voisins pour obtenir la liste des Proxy, invoque les méthodes sur un Proxy.

Les concepts de base de JINI sont :

- **La découverte** : c'est un processus par lequel un client trouve un service lookup pour enregistrer ou demander des services.
- **L'enregistrement d'un service (join)** : Lorsqu'un client reçoit le Proxy du lookup sur lequel il souhaite s'enregistrer, il le fait via la méthode

«register()» de l'interface «ServiceRegistrar», de cette façon, le service client a la capacité de transmettre au lookup son propre proxy.

- **La recherche d'un service (lookup)** : La recherche d'un service se déroule de la même façon que dans l'enregistrement, en invoquant la méthode « lookup() » sur le Proxy du LUS que nous avons reçu lors de la phase de découverte. La spécification de Jini n'impose aucun protocole pour la communication entre les services, bien que l'implémentation de Sun utilise RMI pour contacter le service lookup.

L'inconvénient de Jini est qu'elle a besoin d'un LUS, sorte d'annuaire centralisé et aussi elle est assez lourde à mettre en place.

### 1.4.2 SLP : Service Location Protocol

SLP est un standard proposé par IETF (Internet Engineering Task Force) pour découvrir spontanément les services dans les réseaux IP. Son architecture est similaire à celle de Jini. Il définit trois types d'agents : Agent Utilisateur (User Agent : UA), Agent de Service (Service Agent : SA) et Agent Annuaire (Discovery Agent : DA). Les UA découvrent les services qui sont offerts par les SA et qui sont enregistrés aux DA. SLP peut aussi tolérer l'absence de DA, en permettant aux SA de diffuser en Multicast les requêtes de recherche de services. Grâce aux DA et l'utilisation soignée de messages Multicast, il peut être utilisé dans les réseaux de grande taille comme les réseaux d'entreprises.

### 1.4.3 UPnP : Universal Plug and Play

*Universal Plug and Play* (UPnP) est un nouveau standard de communication entre périphériques, pour les petites entreprises ou les réseaux résidentiels (SOHO - *Small Office Home Office*), développé par un consortium dont fait partie *Intel* et *Microsoft*. Il a été développé dans l'optique d'être un standard convivial et flexible pour des réseaux ad-hoc ou non gérés pour des résidences ou des petits bureaux, des endroits publics. UPnP est plus qu'une simple extension du standard de périphériques Plug and Play. Il permet aux différents périphériques d'un réseau de se configurer automatiquement, d'offrir des services de façon dynamique et transparente[13].

UPnP est une solution qui comporte aussi quelques désavantages. Premièrement, une modification est nécessaire au sein des applications afin de permettre aux applications de communiquer avec les dispositifs UPnP. Deuxièmement, les périphériques tels que les pare-feu et les routeurs NAT doivent être remplacés par des dispositifs équivalents supportant UPnP. Finalement, UPnP peut créer une brèche dans un réseau. En effet, tous les programmes et les dispositifs UPnP peuvent ouvrir des trous d'épingle sans authentification.

En résumé, un programme illicite pourrait ouvrir des portes sur un pare-feu ou un routeur NAT et rendre un réseau vulnérable. Cette solution ne peut donc pas être utilisée dans des réseaux d'entreprises.

## 1.5 Problématique

L'un des principaux problèmes liés au réseau domotique est le problème de l'accès distant aux équipements et dispositifs déployés au niveau du réseau domotique, qui peuvent utiliser des adresses IP et un ensemble de commandes qui ne sont pas nécessairement connus à l'extérieur du domaine local. L'utilisation des pare-feu et des traducteurs d'adresses réseau (NAT) pour protéger l'accès au domaine locale peut compliquer davantage l'exploitation du réseau domotique. La figure 1.4 illustre cet environnement.

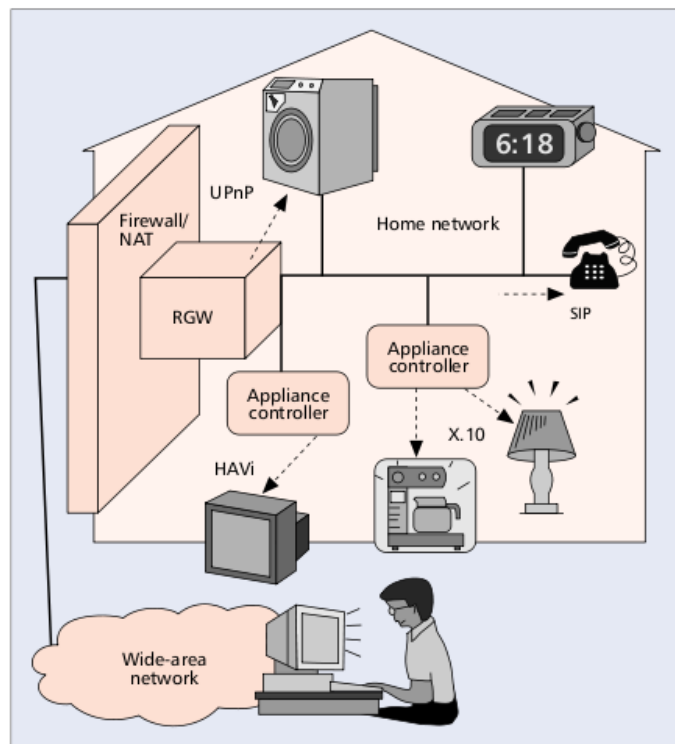


FIG. 1.4 – Problème d'accès dans un réseau domotique

Pour répondre à ces différentes problématiques, nous proposons d'utiliser le protocole SIP comme le protocole adéquat pour le développement de notre plateforme domotique. Dans le chapitre suivant, nous exposerons les critères à partir desquels nous avons choisis SIP.

## 1.6 Conclusion

Ce chapitre a été consacré pour placer le projet dans son cadre général en définissant le terme domotique et en présentant ses avantages. De plus, nous avons

cit  les diff rentes technologies appliqu es dans ce domaine. Cette  tude nous a permis de d celer les probl matiques du domaine ; ces m mes probl matiques que notre projet contribue   r soudre. Dans le chapitre suivant nous pr sentons la sp cification des besoins et des exigences qui permettent de d terminer la technologie appropri e   notre plateforme domotique.

## Chapitre 2

# Spécification des besoins

### 2.1 Introduction

Dans ce chapitre, nous introduisons la première étape de notre processus de développement, à savoir la spécification des besoins. Tout au long de cette étape, nous allons définir la portée et les besoins fonctionnels du système.

### 2.2 Objectifs

L'objectif du projet HomeSIP est de mettre en place une plateforme domotique à l'ENSEIRB (Ecole Nationale Supérieure d'Informatique Electronique et Radiocommunications de Bordeaux) mettant en oeuvre le protocole SIP. Ce projet est par nature un projet orienté embarqué, composé de différents systèmes électroniques sous Linux embarqué.

Le projet consiste au développement d'une plateforme domotique basée sur le protocole SIP. L'idée est donc d'utiliser SIP pour collecter des informations provenant de différents capteurs et de piloter aussi différents équipements (actionneurs).

Le projet HomeSIP consiste donc :

- En une infrastructure matérielle composée de capteurs et d'actionneurs connectés à des systèmes embarqués qui sont communicants et qui possèdent tous une connectivité IP.
- A mettre en place des passerelles entre les appliances domestiques (capteurs filaire et ZigBee, équipements X10,...) et le réseau IP pour permettre aux utilisateurs l'accès au réseau domestique à travers le protocole SIP ; Le déploiement des passerelles nécessite le développement des logiciels adéquats dans les systèmes embarqués sous Linux.
- A développer des nouveaux services autour de la plateforme.

Le projet HomeSIP sera intégré au final à la plateforme TelIP (Téléphonie par IP) en exploitation à l'ENSEIRB. La plateforme TelIP permet des communications VoIP au sein de l'école mais autorise aussi l'accès vers le réseau télépho-

nique classique RTC (Réseau Téléphonique Commuté).

HomeSIP est donc la continuité de TelIP et l'interopérabilité entre les deux plateformes est totale car toutes deux basées sur le protocole SIP, protocole hautement interopérable.

## 2.3 Etude de l'existant

La domotique est une composante du marché du M2M (Machine to Machine). Le M2M est une infrastructure basée autour d'un réseau qui autorise des communications directement entre équipements ou via un serveur central sans aucune intervention humaine. Cela permet par exemple la capture automatique de données et leur traitement par les équipements après transmission. Ce marché émergent est en fait l'évolution « naturelle » de la mise en place de la connectivité réseau et en particulier la connectivité IP dans les équipements électroniques.

Le marché du M2M s'appuie aussi sur les mutations technologiques opérées depuis quelques années :

- Les protocoles Internet sont des standards de fait. Ils sont facilement intégrables dans les équipements électroniques.
- Les fonctionnalités électroniques sont toujours de plus en plus intégrées.
- Les matériels électroniques sont puissants et bon marché : capteurs miniatures MEMS (Micro Electro Mechanical System), tags RFID (Radio Frequency ID).
- Le processeur 32 bits va devenir dans les années qui viennent le processeur de base dans l'embarqué et va être prépondérant en volume par rapport au processeur 8 ou 16 bits. C'est une condition nécessaire pour pouvoir mettre en oeuvre Linux.
- On dispose maintenant de connexions réseaux permanentes à bas prix qui ne sont plus facturées à la durée : réseau ADSL, réseau GPRS.
- Les connexions sans fil sont de plus en plus utilisées. Le coût d'installation est aussi intéressant car l'on minimise le câblage.
- On dispose de standards, synonymes d'interopérabilité et de maîtrise des coûts : normes RFC pour les protocoles Internet, normes UIT (ADSL...), normes ETSI (GSM, GPRS), normes IEEE (Wifi, Zigbee).

Le M2M correspond donc à la convergence électronique-informatique communicantes par Internet. On parle d'Internet diffus ou ambient (ubiquitous Internet) et d'informatique massivement répartie ou informatique diffuse (ubiquitous computing). Selon le cabinet d'étude IDATE[14] qui a publié en juin 2005 une étude récente sur ce sujet, le marché du M2M se compte en milliards d'équipements électroniques et en centaines de milliards d'objets ou capteurs communicants. Selon IDATE, 92 millions de modules ont été vendus en 2004, quelle que soit la connectivité réseau choisie. Ce chiffre devrait atteindre 500 millions de modules d'ici 2010 pour 2 milliards de machines et de 100 milliards d'objets communicants.

Le M2M se développe autour d'industries verticales par secteur d'activité. Nous

pouvons citer comme domaines d'application :

- La gestion de sites et de bâtiments, la domotique.
- La sécurité, la vidéosurveillance, la navigation,...
- Les transports, la gestion de flotte automobile.
- La télémessure, le relevé de compteurs.
- La gestion de l'énergie.
- La télémédecine.
- La monétique.
- Le commerce.

Le projet HomeSIP est par nature un projet de domotique et s'inscrit dans le marché M2M. Actuellement, les périphériques SIP utilisés dans le projet sont des PC, des téléphones SIP ainsi que des PDA. Seules, deux passerelles SIP/réseau de capteurs sont actuellement déployées. Nous serons mener dans ce projet à intégrer les passerelles SIP/X10 pour la commande d'appareils électriques et les passerelles SIP/réseau de capteur ZigBee.

La passerelle SIP/réseau de capteurs existante est composée de matériels du commerce pour ne pas avoir des développements hardware à réaliser. Elle se compose :

- Des cartes ARM9 d'Eukréa. Cette carte est présentée en détail dans[15]. Son grand avantage est qu'elle supporte bien sûr Linux mais qu'elle possède de nombreuses E/S pour pouvoir connecter différents capteurs et actionneurs tel que les liaisons série et les bus USB. La carte possède aussi une interface réseau pour pouvoir remplir son rôle de passerelle.
- Des capteurs de température iButton DS1920 de Dallas Semiconductor avec une interface pour les connecter à une liaison série. L'interface de pilotage d'un iButton étant standard, il est possible d'en rajouter d'autres[16].

SIP est un protocole mettant en jeu la gestion des sessions mais ne gère pas les commandes que l'on veut passer. Nous souhaitons dialoguer entre serveur et clients SIP, pour cela nous avons besoin d'établir une norme. Cette norme sera établi en utilisant SOAP pour la commande des différents dispositifs.

## 2.4 La spécification des exigences

La pertinence du produit réalisé ne peut avoir de sens sans une spécification préalable des besoins et des exigences. Pour rendre pleinement les services attendus, notre plateforme domotique doivent respecter un certain nombre de contraintes. Plus globalement, la plateforme doit :

- Permettre une grande hétérogénéité.
- Tolérer des comportements très dynamiques
- Fonctionner en l'absence d'administrateur.

### 2.4.1 Hétérogénéité

Alors que les réseaux informatiques classiques sont formés de composants similaires (principalement des ordinateurs) comparables en termes de puissance,

de stockage, de mémoire et de bande passante, les réseaux domotiques sont composés de dispositifs hétérogènes. Ils comportent aussi bien du matériel Hi-Fi que des dispositifs mobiles et du matériel informatique. Une autre différence porte sur l'hétérogénéité des moyens de communication.

La solution développée doit permettre l'interfonctionnement entre les différentes technologies domotiques (UPnP, HAVI, X10,...) qui doit être transparent à l'utilisateur final. Dans cette perspective, le protocole SIP peut assurer l'interfacage entre les technologies domotiques via des passerelles que nous devrons développer. En plus, puisque SIP est basé sur IP, il peut être mis en application en utilisant n'importe quel langage de programmation, sur n'importe quel système d'exploitation ou plateforme matériel.

### 2.4.2 Nommage et adressage

- Pour les appliances qui ne possèdent pas une connectivité IP, une passerelle doit être utilisée pour fournir l'interfonctionnement entre le réseau IP et les dispositifs domotiques.
- Les appliances doivent avoir des adresses dans un format générique qui peut être utilisé par toute entité dans le but de communiquer avec les dispositifs domotiques indépendamment de leur type d'adressage.

### 2.4.3 La mobilité

Les réseaux domotiques sont particulièrement dynamiques, et ce à plusieurs titres. Tout d'abord, un réseau domotique donné évolue dans le temps en fonction des dispositifs qui le constituent. Il commence avec le premier appareil communicant que l'utilisateur achète et évolue au gré des achats, des ventes, des pannes et des pertes.

Un réseau domotique est aussi physiquement dynamique. Les éléments qui le composent ne sont pas interconnectés en permanence : les appareils peuvent être mobiles, sous tension ou non ; les canaux de communication peuvent être bruités ou temporairement indisponibles.

Plus généralement, aucune supposition ne peut être faite sur la disponibilité d'un dispositif dans le réseau à un instant donné. Par conséquent, aucun constituant du plateforme à développée ne doit être indispensable au fonctionnement global. C'est une grande différence par rapport aux réseaux informatiques classiques, pour lesquels on peut raisonnablement estimer que certains des dispositifs sont disponibles en permanence.

La mobilité implique aussi le problème de la gestion de la localisation. Le protocole SIP permet de localiser des équipements au sein d'un réseau à partir de l'enregistrement des passerelles auprès d'un serveur SIP qui est en mesure de les localiser. Par exemple, une passerelle s'enregistre à un serveur SIP en effectuant une requête. La localisation de la passerelle étant connue, différentes sessions offrant divers services pourront être ouvertes avec cette passerelle. En effet, SIP permet à un usager, indépendamment de sa localisation d'être accessible dans un réseau.

#### 2.4.4 La fiabilité

- Le protocole de communication doit fournir une flexibilité qui permet de supporter différents types de données («payload») et de transporter des commandes et des réponses de différents dispositifs gérés en réseau. SIP est un protocole qui a la caractéristique d'être extensible. Cette extensibilité lui permet de transporter une charge utile arbitraire. En effet, SIP peut être combiné à d'autres protocoles de communication, codecs ou encore formats de fichiers afin d'offrir des nouveaux services au travers des sessions SIP.
- Il doit y avoir une séparation entre la couche transport et les types des données échangés.
- Puisque la plupart des communications avec les appareils gérés en réseau doivent être effectuées en temps réel, le protocole doit supporter la transmission des messages de commande.
- La notification d'évènement est très importante pour l'interaction avec les appareils gérés en réseau (par exemple avis que votre machine à laver soit au-dessus d'une certaine température), par conséquent, le protocole doit supporter la souscription à des évènements et les notifications.
- Le protocole doit être capable d'encapsuler de diverses caractéristiques d'appareils. Nous utiliserons le modèle de données SOAP pour représenter les caractéristiques des différents appliances déployés dans le réseau domotique.

#### 2.4.5 Le mode de communication

Il y a différents modes pour agir avec les appareils gérés en réseau, Les modes d'interaction suivants sont exigés :

- **Contrôle** : il doit être possible d'envoyer des commandes aux appareils, par exemple, placer la température de climatiseur à 20 degrés.
- **Requêtes** : le protocole doit assurer la demande de l'état d'appareils gérés en réseau, par exemple, quelle est la température d'un capteur de température iButton.
- **Avis d'évènement** : Le protocole doit permettre à des utilisateurs de souscrire pour recevoir des notifications quand des évènements se produisent, par exemple, informer le mobile d'un utilisateur quand l'alarme de la maison est placée hors fonction.
- **Découverte** : le protocole doit garantir à des utilisateurs de rechercher des appareils pour répondre à des exigences particulières, par exemple, trouver un appareil qui permet de faire une tasse du café.

#### 2.4.6 Absence d'administrateur

Enfin, les utilisateurs des réseaux domestiques sont très différents de ceux des réseaux informatiques. L'utilisateur d'un réseau informatique a souvent bénéficié d'une formation adaptée. Les appareils sont pour lui des outils de travail

et il est prêt à suivre quelques procédures fastidieuses pour les maintenir en fonction. Pour les opérations complexes, un administrateur possède le niveau d'expertise adapté et des ressources dédiées.

A contrario, l'utilisateur d'un réseau domotique interagit avec des objets du quotidien. Il n'a généralement bénéficié d'aucune formation particulière et peut tout ignorer du fonctionnement d'un réseau. Quand bien même il disposerait de compétences adaptées, il ne consacrerait probablement pas les ressources nécessaires pour configurer, administrer, et superviser régulièrement le sien. Les réseaux domotiques présentent donc des différences essentielles par rapport aux réseaux informatiques traditionnels, résumées dans le tableau ci-dessous.

Réseaux informatiques	Réseaux domotiques
Dispositifs similaires.	Dispositifs hétérogènes.
Moyens de communication rationalisés.	Moyens de communication hétérogènes.
Interconnexion supposée permanente des dispositifs.	Interconnexion erratique des dispositifs.
Utilisateurs formés et actifs.	Utilisateurs non formés et passifs.
Évolutions du réseau rare et maîtrisées.	Évolutions du réseau fréquentes.
Administrateurs.	Pas d'administrateurs.

## 2.5 Les besoins fonctionnels

Dans ce paragraphe nous exposerons l'ensemble des besoins auxquels doit répondre notre système domotique. En effet, étant donné la nature de notre application qui obéit à une architecture client/serveur, il s'agit de développer un système qui permet d'utiliser et d'administrer la plateforme domotique soit localement par PC, soit à distance par un mobile SIP.

L'administration sera appliquée aux différents dispositifs domotiques. En effet, notre système devra être capable :

- De découvrir les dispositifs et de leurs services.
- De fournir une description détaillée de chaque dispositif et de ses services disponibles.
- De visualiser les actions que l'utilisateur peut invoquer.
- De permettre à l'utilisateur de savoir les états actuels de ces dispositifs.
- De permettre à l'utilisateur d'être notifié sur les différents changements dans son réseau.
- De garantir à l'utilisateur l'activation et la désactivation des différents dispositifs.
- De permettre de commander les dispositifs sur le réseau.

## 2.6 Structure et solution retenue

En se basant sur ce qui précède, une solution qui semble conforme aux besoins exprimée et aux objectifs déjà fixés se résume dans un système domotique illustré par la figure 2.1.

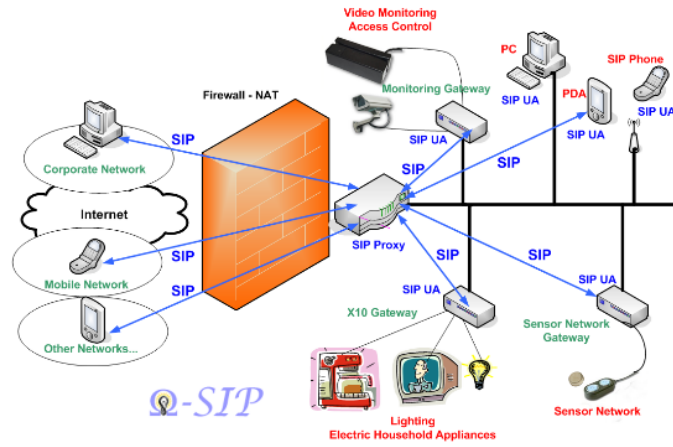


FIG. 2.1 – Plateforme HomeSIP

La plateforme domotique se compose des entités suivantes :

- **Les dispositifs** : ce sont des dispositifs domotiques non IP gérés localement et à distance. Ils peuvent être des dispositifs X10, des caméras de surveillances ou des capteurs de température.
- **Les Passerelles SIP** : il gère les interactions entre les équipements domotique et le monde IP.
- **L’infrastructure SIP** : elle est composée de Proxy SIP, Register SIP et des User Agent SIP

La plateforme matérielle est basée sur l’architecture SIP de l’ENSEIRB. Un routeur central assure la redistribution sur le réseau Ethernet de l’école suivant le protocole SIP. Le réseau Ethernet de l’ENSEIRB sert d’ossature à la plateforme HomeSIP. Toutes les communications utilisent ce protocole. Par l’intermédiaire de différentes passerelles, il est ainsi possible d’avoir accès à certains éléments basés sur X10, technologie très répandue en domotique pour la commande d’appareils par courant porteur. Il pourra être aussi possible d’accéder à différents capteurs de température ou encore contrôler la vidéo surveillance.

## 2.7 Conclusion

L’étape de spécification des besoins nous a permis d’avoir une idée claire sur la faisabilité de notre application à travers la détermination des principales fonctionnalités de notre système; cette étape s’avère très importante dans notre processus de développement vu que le reste du projet ne sera autre que d’essayer de satisfaire les besoins des utilisateurs exprimés précédemment.

Dans le chapitre suivant, nous nous focaliserons sur l’importance de l’utilisation du protocole SIP dans un environnement domotique.

# Chapitre 3

## Le protocole SIP

### 3.1 Introduction

Dans ce chapitre, nous détaillons le protocole SIP. Dans la partie une, nous présentons la structure générale de ce protocole. Ensuite, nous étudions dans la deuxième partie l'architecture protocolaire et les différents types de messages SIP utilisés. Enfin, nous terminons par une comparaison entre SIP et d'autre protocole pour une utilisation dans un environnement domotique.

### 3.2 Structure générale du protocole SIP

SIP est un protocole de signalisation de bout en bout permettant d'établir une session entre deux équipements pour un échange de données (ou d'un flux) par Internet. Il est le descendant des protocoles de signalisation plus classiques comme SS7 (Système de Signalisation numéro 7) de l'UIT-T (Union Internationale des Télécommunications) que l'on retrouve dans le RTC (Réseau Téléphonique Commuté), le RNIS (Réseau Numérique à Intégration de Services), le GSM. Il s'en distingue radicalement par sa flexibilité et surtout par le fait qu'il n'est pas lié à un type de données à échanger et encore moins à un type de réseau de transport. De plus il peut être étendu et s'adapter aux évolutions futures.

Cette qualité est sa grande force. On peut l'utiliser pour mettre en relation 2 équipements pour tout type d'échange de données :

- Voix sur IP. C'est l'échange de données le plus connu.
- Messagerie instantanée IM (Instant Messaging).
- Vidéo

Il existe plusieurs RFC (Request For Comments) disponibles en ligne [19] qui correspondent au protocole SIP telque :

- RFC 3265 : Session Initiation Protocol (SIP)-Specific Event Notification.
- RFC 3428 : Session Initiation Protocol (SIP)-Extension for Instant Messaging.

### 3.3 Architecture du protocole SIP

Le protocole est bâti sur une architecture Client/Serveur et utilise des messages textuels. Les messages sont transportés par les protocoles de transport réseaux TCP ou UDP. Le message possède un en-tête et un corps. L'en-tête définit les paramètres nécessaires au routage du message et à l'établissement de la session. Le corps définit les caractéristiques de la session à l'aide d'un protocole de description de session. Les parties établissement de session et description de session sont séparées. La figure 3.1 donne le positionnement de SIP dans la constellation des protocoles Internet.

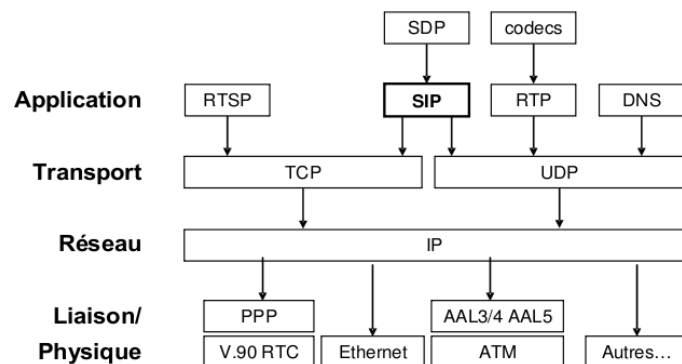


FIG. 3.1 – Positionnement du protocole SIP parmi les protocoles Internet

SIP reprend des éléments techniques des protocoles SMTP (Simple Mail Transport Protocol) et HTTP (Hyper Text Transport Protocol) ; Ceci veut dire que :

- SIP est basé sur le concept d'application client/serveur. On a une application d'extrémité cliente qui désire se mettre en relation avec une application d'extrémité serveur. SIP possède un numéro de port réservé : le 5060.
- SIP est un protocole de type commande/réponse. Ces commandes/réponses sont structurées sous la forme de chaînes de caractères ASCII directement lisibles et ressemblent à celles de HTTP (les codes d'état pour les réponses par exemple).

### 3.4 Les fonctions utilisées par SIP

Pour établir et terminer des communications multimédia, SIP utilise les 5 fonctions suivantes :

- **User location** : permet de localiser le poste terminal utilisé pour communiquer.
- **User capabilities** : détermine quels médias vont être échangés (voix, vidéo et données) ainsi que les paramètres associés.

- **User availability** : détermine si le poste appelé souhaite communiquer et autorise l'appelant à le contacter.
- **Call setup ou " ringing "** : avertit les parties appelant et appelé de la demande d'ouverture de session (sonnerie ou message de réception d'appel) et mise en place des paramètres d'appel.
- **Call handling** : gère le transfert et la fermeture des appels.

Pour établir une communication, l'appelant, désigné par client, adressera sa requête à un serveur SIP, qui lui donnera les moyens de communiquer. Il existe 5 types de serveurs :

- **Les Users Agents** : Il peut être, par exemple, un terminal de téléphonie ou de visioconférence sur IP, un serveur audio ou vidéo ou encore une passerelle vers un autre protocole.
- **Les Serveurs Proxy** : Un serveur proxy a la charge de router les messages SIP. Il a uniquement un rôle dans la signalisation et il ne gère pas de media. Il n'est en général à l'origine d'aucune requête exceptée la requête CANCEL utilisée pour libérer une session.
- **Les Serveurs Redirect** : Les serveurs Redirect aident à localiser les User Agent SIP en fournissant une adresse alternative à laquelle l'utilisateur appelé peut être joint. Le Redirect Server réalise simplement une association (mapping) d'adresses vers une ou plusieurs nouvelles adresses. Ainsi, lorsqu'un client appelle un terminal mobile, l'appel est redirigé vers le Proxy Server le plus proche ou en mode multicast, le message émis est redirigé vers toutes les sorties auxquelles sont reliés les destinataires.
- **Les Serveurs de Localisation** : Lorsqu'une entité SIP souhaite joindre un correspondant à partir de son adresse SIP, elle est renseignée par le Location server qui accède à la base d'information tenue à jour par le serveur Registrar.
- **Les Serveurs Registrar** : Afin de pouvoir joindre une personne à partir de son adresse SIP, une entité dans le réseau doit maintenir une correspondance (mapping) entre les adresses IP et les adresses SIP. C'est le rôle du serveur Registrar. Un utilisateur peut donc changer d'adresse, il lui suffit de s'inscrire auprès du Registrar en lui indiquant son adresse SIP et son adresse de machine sur le réseau.

SIP peut être utilisé pour le contrôle de conférences multimédia, d'appels téléphoniques sur IP et bien d'autres types de communications. Les communications peuvent être en unicast (l'expéditeur envoie un flux au destinataire) ou en multicast (l'expéditeur envoie un flux vers le réseau et le réseau le duplique pour le faire parvenir à tous les destinataires). Les participants sont des utilisateurs finaux, des serveurs de media, des serveurs de signalisation SIP, ou bien des passerelles vers d'autres réseaux. La figure 3.2 donne un exemple d'infrastructure SIP.

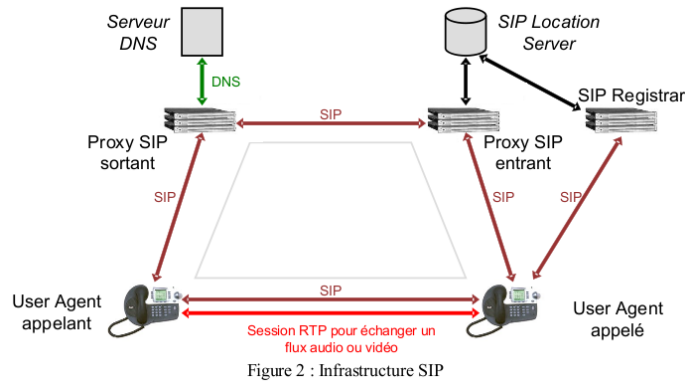


FIG. 3.2 – Infrastructure SIP

## 3.5 Les messages SIP

Un message SIP peut être à la fois une requête d'un client vers un serveur ou une réponse d'un serveur vers un client.

### 3.5.1 Les requêtes de base SIP

Les requêtes de base SIP appelés encore « méthodes », sont au nombre de six. Ces requêtes de base permettent de localiser, d'adresser un élément du réseau et lui transmettre les informations de signalisation[18] :

- **Invite** : ce message est une demande d'établissement de liaison. Le type de session, l'adresse IP, le port, et le type du codec sont inscrits dans le corps du message. L'envoi d'un message « invite » durant une session existante donne lieu à une réinvitation et est utilisé pour la modification des paramètres de la session actuelle.
- **ACK** : termine la demande de liaison (invite). Il est uniquement utilisé pour ceci. Si lors de la demande de liaison le corps du message invite ne contient pas les informations sur le type médias, alors le ACK devra les contenir.
- **Options** : demande à un autre agent ses comptabilités. La réponse contiendra la liste des méthodes qu'il supporte, ces codecs ou d'autres informations. L'agent questionné répondra à ce message comme s'il s'agissait d'un message « Invite ».
- **Bye** : termine une communication. L'agent arrête l'envoi de paquets de type media (RTP).
- **Cancel** : termine une communication en cour d'établissement.
- **Register** : permet à un agent de s'enregistrer ou de mettre à jour sa localisation et sont URL auprès d'un serveur enregistrement, celui-ci pourra à son tour mettre à jour le serveur de localisation, ces données seront utilisées pour la redirection des communications.

### 3.5.2 Les autres requêtes SIP

Afin d'étendre les possibilités de SIP, des nouvelles méthodes ont été ajoutées. Actuellement on peut en compter 8 mais il est vraisemblable que cette liste va s'agrandir au fil du temps :

- **Info** : est utilisé pour transmettre de la signalisation.
- **Prack** : ce message est une confirmation à un message de réponse temporaire.
- **Refer** : permet à un agent de demander à un autre agent d'exécuter une requête particulière.
- **Comet** : est utilisé pour annoncer à un agent qu'il doit avertir l'utilisateur que certaines conditions (ex. QoS ) ont été réunies.
- **Subscribe** : permet de s'inscrire de façon à être informé lors de l'exécution d'un événement donné.
- **Unsubscribe** : annule une inscription préalable.
- **Notify** : est utilisé pour informer les utilisateurs inscrits qu'un événement a eu lieu.
- **Message** : permet l'envoi d'un message vers un utilisateur. Le corps du message peut être de type HTML, texte ou autre.

### 3.5.3 Les réponses SIP

Elles permettent à un élément du réseau de répondre à une requête. Elles ont été créées sur le modèle des réponses HTTP. Ces réponses sont divisées en six classes, cela permet de connaître rapidement le type de réponse [19].

1. Information, 1xx par exemple : 100 en cours d'essai (trying), 180 sonne (ringing).
2. Succès, 2xx par exemple : 200 OK.
3. Redirection, 3xx.
4. Erreur du client, 4xx par exemple : 400 mauvaise demande.
5. Erreur du serveur, 5xx par exemple : 503 services non disponibles.
6. Erreur générale, 6xx par exemple : 600 occupé, 604 n'existe pas.

### 3.5.4 Format des messages SIP

Un message SIP est composé de deux parties : un Header et un Body séparés par une ligne vide. Le Header d'un message SIP contient des champs tels que *To*, *From*, *Via*, *Contact*, les champs utilisés pour introduire la source, la destination et le chemin suivi par une requête SIP. La figure 3.3 présente un enchaînement typique pour établir une communication VoIP par SIP.

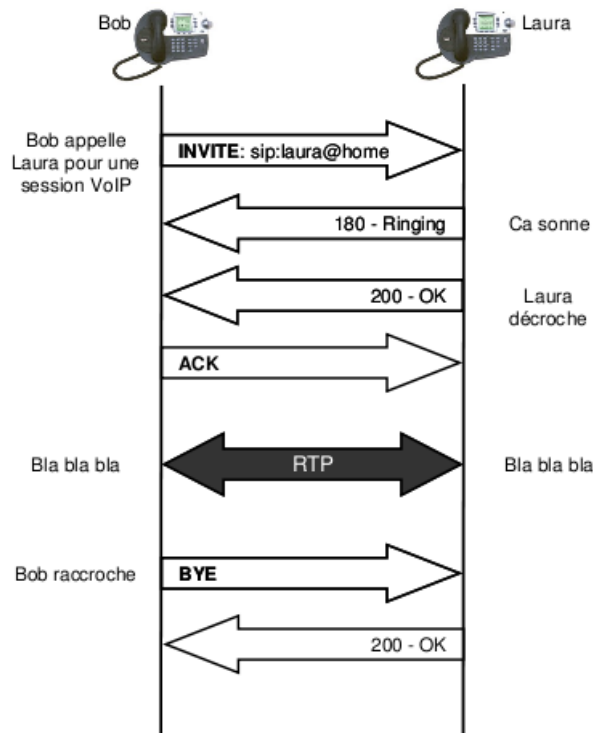


FIG. 3.3 – Échange des messages SIP pour une communication VoIP

### 3.6 Le protocole SIP et la domotique

Le protocole SIP possède des qualités intrinsèques pour la domotique :

- Il supporte un mode d’adressage abstrait.
- Il apporte un niveau de confidentialité et de sécurité. Les données échangées peuvent être authentifiées et chiffrées.
- Il supporte différents flux d’échange et différents mécanismes de communication.
- Il supporte tout type de charge (payload) dans ses messages en utilisant le format MIME.
- Il peut s’interfacer à d’autres technologies domotiques via des passerelles : bus CAN (Control Area Network), courant porteur (norme domotique X10).
- Il supporte la mobilité.
- Il réutilise l’infrastructure SIP classique.
- SIP est un protocole rapide. La séparation entre ses champs d’en-tête et son corps du message facilite le traitement des messages et diminue leur temps de transition dans le réseau.

En domotique, on distingue 2 types de transferts de données :

**Le transfert synchrone** : Une donnée actuelle est acquise par une application, par exemple la valeur courante d'un capteur. On peut vouloir aussi réaliser une action, par exemple changer l'état courant d'un actionneur. Un capteur peut être un capteur de température, un actionneur, une gâchette électrique de porte. Nous devons pouvoir réaliser des actions «GET input» et «PUT output».

**Le transfert asynchrone** : Une alarme est envoyée de façon asynchrone à une application en cas de survenue d'un problème quelconque. Un exemple d'alarme est la détection d'un début d'incendie. nous devons pouvoir collecter des alarmes ou «TRAP».

Suivant les RFC 3261, 3265 et 3428, nous pouvons utiliser SIP pour assurer ces 2 type de transferts :

- Le message SIP MESSAGE pour des actions PUT ou GET. Ce message a été originellement introduit pour la messagerie instantanée (RFC 3428).
- Les messages SIP SUBSCRIBE et NOTIFY les évènements de type TRAP. Ces messages sont originellement utilisés pour la notification de présence (RFC 3265).

La figure 3.4 présente l'utilisation du protocole SIP dans un environnement domotique.

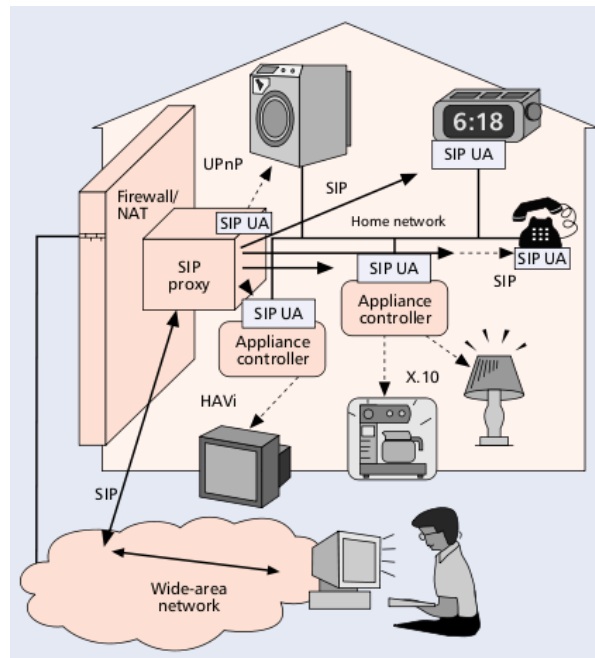


FIG. 3.4 – Utilisation de SIP dans un environnement domotique

### 3.6.1 Comparaison SIP / HTTP

Le HTTP (*Hypertext Transfer Protocol*) est un protocole de communication client-serveur développé pour le Web. Une passerelle SIP pourrait être considérée comme un serveur web, avec les services et les applications qu'elle offre, par analogie aux pages Web. Toutefois, la passerelle pourrait être mobile et peut générer des notifications asynchrones. HTTP n'offre pas un bon support pour la mobilité et les notifications. En outre, HTTP fonctionne sur TCP, et une pile TCP est plus grande et plus complexe qu'une pile UDP ce qui ne correspond pas aux contraintes des systèmes embarqués dans un environnement domotique.

### 3.6.2 Comparaison SIP / SNMP

SNMP (*Simple Network Management Protocol*) est un protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, superviser et de diagnostiquer des problèmes réseaux, matériels à distance.

SNMP offre les mêmes possibilités d'échanges que SIP dans un contexte domotique. Il possède :

- Les messages SNMP PUT et GET pour des actions PUT et GET.
- Le message SNMP TRAP les événements de type TRAP.

Mais SNMP possède des défauts majeurs par rapport à SIP :

- SNMP est un protocole complexe malgré son nom.
- Il est difficile à utiliser. L'extension d'un agent SNMP est compliquée.
- SNMP n'est pas un protocole très sécurisé surtout dans sa version 1.
- Un agent SNMP n'a pas une empreinte mémoire faible, ce qui est un handicap pour un système embarqué avec peu de mémoire.
- Le protocole SNMP n'a pas d'applications clientes populaires. SIP en a de nombreuses (gaim par exemple).

Le protocole SIP s'impose de lui-même dans un contexte domotique et de convergence électronique-informatique-réseaux.

### 3.6.3 Comparaison SIP / SMTP

SMTP (*Simple Mail Transfer Protocol*) est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique. SMTP peut être utilisé en tant que protocole de notre plateforme domotique car il supporte un niveau d'adressage abstrait, il supporte tout type de charge et il supporte la mobilité. Toutefois, SMTP ne prend pas en charge les événements et les sessions. En plus, SMTP peut présenter des temps de latence importante lors du transfert des messages ce qui ne correspond pas à un environnement domotique où le transfert doit être en temps réel.

## 3.7 Conclusion

Dans ce chapitre, nous avons présenté le protocole SIP et exposé les arguments qui plaident pour son utilisation dans notre plateforme domotique par rapport à d'autre type de protocole tel que SNMP ou HTTP. Dans le prochain chapitre, nous développons la partie conception de la base de données et de l'application.

# Chapitre 4

## Conception de l'application

### 4.1 Introduction

Une fois l'analyse des besoins et la spécification des exigences du projet sont élaborées, nous abordons la partie conception qui constitue la phase la plus importante dans le cycle de développement d'un projet puisqu'elle permet de traduire l'ensemble des exigences exposées en une solution.

Dans le cas de notre application, nous tenons à expliquer la conception de notre système domotique en étudiant les différentes interactions entre l'ensemble des entités du système.

### 4.2 Méthodes de conception

Une méthode de conception définit une démarche reproductible qui vise l'obtention des résultats fiables. D'une manière générale, les méthodes permettent de construire des modèles à partir d'éléments de modélisation qui constitue des concepts fondamentaux pour la représentation des systèmes ou des phénomènes. Les méthodes définissent également une représentation souvent graphique qui permet d'une part de manipuler aisément les modèles, et d'autre part de communiquer et d'échanger l'information entre les différents intervenants. Une bonne représentation recherche l'équilibre entre la densité d'information et la lisibilité [20].

#### 4.2.1 Méthode fonctionnelle

Elle consiste à définir les fonctions des composantes d'un système et leurs relations fonctionnelles. Le système est conçu d'un point de vue fonctionnel, en partant d'une vue de haut niveau, qu'on affine successivement afin d'obtenir une conception plus détaillée. L'état du système est centralisé et partagé par les fonctions qui agissent sur cet état.

## 4.2.2 Méthode orientée objets

C'est une méthode de conception qui permet de programmer en terme d'objets. Le système est vu comme un ensemble d'objets, plutôt que comme un ensemble de fonctions. L'état du système est centralisé, et chaque objet gère l'information concernant son propre état. Les objets disposent d'un ensemble d'attributs qui définissent leurs états, et d'un ensemble d'opérations qui permettent d'agir sur ces attributs [21].

## 4.3 Conception de la base de données

MERISE est une méthode de conception, de développement et de réalisation de projets Informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

La méthode Merise propose une méthode de conception et de développement de systèmes d'information complète, détaillée, en grande partie formalisée, qui garantit une informatisation réussie. La séparation des données et des traitements en plusieurs modèles conceptuels et physiques garantie la stabilité d'agencement des données, la possibilité de remaniement des traitements et la longévité du système.

La méthode Merise prévoit une conception par niveaux, et définit pour cela 3 niveaux essentiels :

- **L'analyse** : Il décrit l'ensemble des données du système d'information, sans tenir compte de l'implémentation informatique de ces données.
- **Modèle conceptuel des données** : Il prend en considération l'implémentation du système d'information par un SGBD (Système de gestion de base de donnée). Ce niveau introduit la notion des tables logiques, et constitue donc le premier pas vers les tables des SGBD.
- **Modèle logique des données** : Il contient finalement les tables définies à l'aide d'un SGBD spécifique.

### 4.3.1 Le niveau conceptuel : Analyse

Notre projet consiste à recueillir des informations, issues de différent type de capteurs et d'actionneur, à distance à l'aide d'une communication IP. Dans notre plateforme, nous allons utiliser des capteurs iButton, des module Zigbee et X10. Dans ce cadre, pour tout type d'équipements nous avons besoin de savoir :

- Le nom de l'équipement.
- Le type de l'équipement, qui peut être un capteur iButton, un module ZigBee ou un dispositifs X10.
- Le port sur lequel est connecté l'équipement (port série /dev/ttyS0,/dev/ttyS1, ou port USB /dev/ttyUSB0).

- L'état de l'équipement (active, désactive).

En plus de ces informations communes à l'ensemble des équipements, nous avons décidé de stocker des informations spécifiques aux différents types de dispositifs manipulés dans notre plateforme.

### Capteur iButton

Pour les dispositifs iButton, nous disposons des capteurs de température ou d'humidité permettant une mesure et un enregistrement facile de la température ou de l'humidité. Pour les capteurs de température nous pouvons énumérer les catégories suivantes :

- iButton DS1920.
- iButton DS1921L.
- iButton DS1921G.

### Dispositifs X10

Notre plateforme domotique fournit les interfaces permettant aux utilisateurs de pouvoir exécuter un certain nombre de fonctions. Grâce à de telles fonctions, l'utilisateur peut envoyer des commandes d'activation et de désactivation à des modules X10. Il peut ainsi commander des équipements tels que l'éclairage ou d'autres appareils électroménagers.

X10 est un protocole qui permet à l'utilisateur d'envoyer des commandes et autres données via les lignes secteur déjà présentes. Ainsi, grâce à un émetteur X10, il est possible d'émettre des commandes de mise en marche ou à l'arrêt vers des récepteurs X10 (des modules d'éclairage X10, des modules d'autres équipements électroménagers X10) raccordés aux prises secteur situées dans les locaux.

Chaque dispositif X10 possède deux codes qui sont utilisés pour son identification. Ceux-ci sont nommés "code site" et "code dispositif" et sont généralement définis grâce au réglage des roues codeuses situées sur le dispositif X10. Dans la figure 4.1 le dispositif X10 est paramétré sur "site A", "dispositif 3".

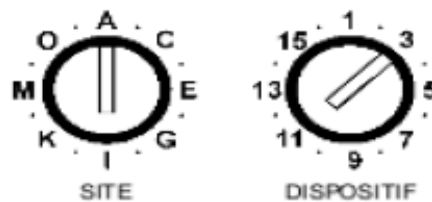


FIG. 4.1 – Roues codeuses pour identification d'un dispositif X10

## Module XBee

Le ZigBee est une norme de réseau sans fil prévue pour les appareils qui demandent peu de débit. La norme ne permet que 250 kilobits/s. Par contre, la portée est correcte (100 m) et la consommation est très faible. Le ZigBee est destiné en priorité à des applications domotiques ou à des systèmes de surveillances simples telque les détecteurs de fumées.

Les modules XBee sont des petits dispositif qui permet d'étendre un réseau ZigBee. Ils se branchent simplement sur une prise de courant et intègre des capteurs de température et de lumière. Le protocole 802.15.4 utilisé par les module Xbee ajoute au paquet de données transmises, une adresse de la source et une adresse du destinataire. Une adresse Xbee est donné par des registres du module. Cette adresse est un numéro de série unique donné en usine par le constructeur.

## Passerelle SIP

Pour piloter à distance notre installation domotique, nous avons besoin d'une passerelle entre l'infrastructure SIP et les dispositifs. Une passerelle SIP/réseau de capteur possède les caractéristiques suivantes :

- Adresse IP.
- Nom de la passerelle.
- URI de la passerelle (adresse SIP).

Notre plateforme met en oeuvre des interactions entre les passerelles SIP et les éléments suivants :

- Les User Agents désignent les agents que l'on retrouve dans les téléphones SIP, les softphones (logiciels de téléphonie sur IP) des PC et PDA. En théorie, on peut établir des sessions directement entre deux User Agents, deux téléphones par exemple. Mais cela nécessite de connaître l'adresse IP du destinataire. Cela n'est pas l'idéal car une adresse IP peut ne pas être publique (derrière un NAT) ou changer et elle est bien plus compliquée à retenir qu'une URI. Les User Agents peuvent donc s'enregistrer auprès des Registrars pour signaler leur emplacement courant.
- Le Registrar est un serveur qui gère les requêtes REGISTER envoyées par les Users Agents pour signaler leur emplacement courant. Ces requêtes contiennent donc une adresse IP, associée à une URI, qui seront stockées dans une base de données spécifique au registrar. Généralement, des mécanismes d'authentification permettent d'éviter que quiconque puisse s'enregistrer avec n'importe quelle URI.

Dans une infrastructure SIP, un Proxy SIP se contente de relayer uniquement les messages SIP pour établir, contrôler et terminer la session. Une fois la session établie, les données, par exemple un flux RTP pour la VOIP, ne transitent pas par le serveur Proxy. Elles sont échangées directement entre les User Agents. Un proxy SIP peut être lui aussi un serveur d'enregistrement pour les Agents SIP.

### 4.3.2 Modèle conceptuel de donnée (MCD)

#### Introduction

Le Modèle Conceptuel des Données (MCD) est la représentation simplifiée de l'ensemble des données manipulées par le système d'information. L'intérêt de ce modèle est d'identifier la signification et la description de chaque information indépendamment de leur organisation et de leur implantation géographique. En effet, les concepts qui apparaissent et qui sont utilisés dans le MCD sont très variables dans le temps et, à quelques exceptions près, constitue une image très fidèle des systèmes d'information futurs. Les différents concepts utilisés dans le MCD sont :

1. **Entité** : C'est une information majeure manipulée par l'organisme et dotée d'une existence propre et identifiable. Elle est présentée par un rectangle où figure son identifiant et ses attributs.
2. **Propriété** : C'est la modélisation d'une information élémentaire présente dans le discours. C'est l'élément descriptif de l'entité ou de la relation.
3. **Identifiant** : C'est une propriété telle que, une valeur de l'identifiant, correspond à une seule occurrence de l'entité. Elle sert à référencier chaque occurrence de l'entité de façon unique.
4. **Relation** : C'est un formalisme qui définit un lien sémantique entre les entités manipulées, elle peut être soit porteuse de données, soit non porteuse de données. On appelle dimension le nombre d'individus composant la relation. Dans le graphe du MCD, elle est représentée par un cercle portant son nom (un verbe) et les propriétés qu'elle porte.
5. **Cardinalité** : Les cardinalités d'une relation entre entités indiquent le nombre d'occurrence maximal et minimal de participation des entités dans la relation. Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

#### Construction du schéma conceptuel

Nous pouvons maintenant définir les types d'entités et les associations de notre base de données en nous inspirant de la spécification des besoins en données.

**Liste des entités** : 7 types d'entités peuvent être identifiés :

- Un type d'entité *User\_Agent* qui possède les propriétés : adresse IP,URI (adresse SIP),le nom et le type de l'agent.
- Un type d'entité *Registrar* qui a comme propriétés : adresse IP,URI et le nom du registrar.
- Un type d'entité *Device* qui possède des informations communes de tout les dispositifs domestiques et ayant comme propriétés : l'adresse IP, le nom

de l'équipement, son type, l'état courante et le port sur lequel l'équipement est connecté.

- Un type d'entité *Passerelle* avec les propriétés :adresse IP,URI, le type et le nom de la passerelle.
- Un type d'entité *Ibutton\_Device* ayant comme propriétés : type et catégorie du capteur.
- Un type d'entité *X10\_Device* ayant comme propriétés : code site et code dispositif.
- Un type d'entité *Xbee\_Module* ayant comme propriété : adresse Xbee.

**Liste des relations :** Nous pouvons distinguer les associations suivantes :

- L'association *Enregistrement\_User* entre *User\_Agent* et *Registrar*.
- L'association *Enregistrement\_Gateway* entre *Passerelle* et *Registrar*.
- L'association *Commande* entre *User\_Agent* et *Passerelle*.
- L'association *Associe* entre *Passerelle* et *Device*.
- L'association *Est\_de\_type* entre *Device* et *Ibutton\_Device*.
- L'association *Est\_de\_type* entre *Device* et *X10\_Device*.
- L'association *Est\_de\_type* entre *Device* et *Xbee\_Device*.

#### **Le modèle conceptuel des données**

- Un utilisateur s'enregistre à un seul Registrar ; Une entité *User\_Agent* est associée via la relation *Enregistrement\_User* à exactement une entité *Registrar*. Un register peut recevoir des requêtes d'enregistrement de plusieurs utilisateurs ; Une entité Registrar est associé via *Enregistrement\_User* à plusieurs entités de type *User\_Agent*.
- Une entité *Passerelle* est associée via la relation *Enregistrement\_Gateway* à exactement une entité *Registrar*. Une entité *Registrar* est associé via *Enregistrement\_Gateway* à plusieurs entités de type *Passerelle*.
- Un utilisateur peut commander plusieurs passerelle SIP/réseau de capteur dans notre plateforme domotique ; Une entité *User\_Agent* est associé via *Commande* à plusieurs entités de type *Passerelle*. Une passerelle peut recevoir des requêtes de commandes de plusieurs utilisateur ; une entité *Passerelle* est associé via *Commande* à plusieurs entités de type *User\_Agent*.
- Une passerelle est liée à plusieurs type d'équipements domestique ; Une entité *Passerelle* est associé via *Associe* à plusieurs entités de type *Device*. Un dispositif est associé à une seul passerelle SIP/réseau de capteur.
- Un équipement domotique peut etre de type X10, iButton ou un module Xbee ; Une entité *Device* peut être associé via *Est\_de\_type* à une entité *X10\_Device*, *Ibutton\_Device* ou *Xbee\_Module*.

#### **4.3.3 Modèle Logique de données**

Dans le modèle logique de donnée les entités se transforment en relations, les propriétés deviennent des attribues et l'identifiant d'une entité devient la clé

primaire de la relation. Les relations permettent de traduire aussi les associations grâce aux clés étrangères.

- L'association binaire *Enregistrement\_User* de type (0-N,1-1) entre les entités *Registrar* et *User\_Agent* se traduit par la redondance de l'identifiant de l'entité *Registrar* dans la relation issue de l'entité *User\_Agent*. Le même raisonnement peut être appliqué à l'association *Enregistrement\_Gateway*.
- L'association binaire *Commande* de type(0-N,0-N) entre les entités *Passerelle* et *User\_Agent* se traduit par la création d'une relation *Commande* qui contient comme attributs les identifiants des 2 entités associées. Ces attributs constituent à eux 2 la clé primaire de la relation et ils sont individuellement clés étrangères.
- L'association binaire *Associe* de type (1-N,1-1) entre les entités *Passerelle* et *Device* se traduit par la redondance de l'identifiant de l'entité *Passerelle* dans la relation issue de l'entité *Device*.
- L'association binaire *Est\_de\_type* de type(0-1,1-1) entre les entités *Device* et *X10\_Device* se traduit par la redondance de l'identifiant de l'entité *Device* dans la relation issue de l'entité *X10\_Device*. Le même raisonnement peut être appliqué pour l'association *Est\_de\_type* entre *Device* et *Ibutton\_Device* et entre *Device* et *Xbee\_Module*.

### Le schéma relationnel

Le schéma relationnel peut être représenté comme suit :

- *User\_Agent* (id\_User, id\_Reg Adresse\_IP\_User, URI\_User, Name, Type\_Agent).
- *Registrar* (id\_Reg, id\_Reg, Adresse\_IP\_reg, URI\_Reg, Name\_Reg).
- *Passerelle* (id\_Pas, Adresse\_IP\_Pas, URI\_Pas, Type\_Pas).
- *Device* (id\_Dev, id\_Pas Adresse\_IP\_Dev, URI\_Dev, Name\_Dev, Type\_Dev, Etat, Port).
- *Ibutton\_Device* (id\_iButton, id\_Dev, Type\_capteur, catégorie).
- *X10\_Device* (Code\_site, Code\_dispositif, id\_Dev).
- *Xbee\_Module* (id\_Xbee, id\_Dev, Adresse\_Xbee).
- *Commande* (id\_User, id\_Pas).

## 4.4 Conception de l'application

Cette étape sera réservée à la conception proprement dite du système et ce en affinant la spécification des composants de l'application et leurs interactions à travers des diagrammes permettant de représenter les interactions entre les éléments du système, en précisant la chronologie des échanges des messages durant l'exécution, et de représenter l'exécution des opérations relatives à une utilisation spécifique du système.

### 4.4.1 Méthodologie adoptée

La méthode UML est devenue le standard industriel de la modélisation. L'UML est sous l'entière responsabilité de l'OMG (Object Management group).

Il se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression des besoins jusqu'à le codage. C'est un langage formel possédant les caractéristiques suivantes :

- Il n'est pas une notation fermée ; elle est extensible, générique et configurable par l'utilisateur.
- Un langage sans ambiguïtés.
- Un moyen de définir la structure d'un programme.
- Une représentation visuelle permettant la communication entre les acteurs d'un même projet.
- Une notation graphique simple, compréhensible même par des non informaticiens.

UML s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel mais, nous allons représenter seulement ceux qui sont utilisés dans notre projet et qui sont :

- Les diagrammes des cas d'utilisations.
- Les diagrammes de séquences.

### Diagramme des cas d'utilisations

Les cas d'utilisation permettent de modéliser et de structurer les interactions entre les utilisateurs au sens large, appelés acteurs et un système.

Les cas d'utilisation représentent un moyen d'analyse des besoins utilisateurs et permettent de relier les actions faites par un utilisateur avec les réactions attendues d'un système. Plus précisément, un cas d'utilisation unitaire est une abstraction d'un ensemble de scénarios concrets effectués sur l'initiative d'un type d'utilisateurs [20].

Les éléments de base des diagrammes des cas d'utilisations sont :

- **Les acteurs** : ils représentent un rôle joué par une entité externe (utilisateurs humain, dispositifs matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut modifier et/ou consulter directement l'état du système, en émettant et/ou en recevant des messages susceptible d'être porteurs de données.
- **Cas d'utilisation** : il représente un ensemble de séquences d'action qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteur/système et apporte une valeur ajoutée à l'acteur concerné.

## Diagrammes de séquences

Un diagramme de séquence montre chronologiquement les interactions entre un ensemble d'objets. Chaque objet dispose d'une ligne de vie (ligne verticale). Sur ces lignes de vie, des périodes d'activités sont indiquées par des rectangles fins qui sont superposés en cas d'appel récursif [20].

### 4.4.2 Représentation des diagrammes des cas d'utilisations

A partir de la définition des besoins, nous identifions les acteurs et leurs interactions avec le système, ce qui permet de déduire assez facilement le diagramme de cas d'utilisation général.

Le diagramme de cas d'utilisation général sera spécifié par un autre cas d'utilisation. Pour décrire la dynamique des cas d'utilisations et les documenter, nous utiliserons les diagrammes de séquences afin d'illustrer les scénarios d'utilisations des différents cas d'utilisations du système ce qui apportera un niveau supérieur de formalisation.

#### Cas d'utilisation général du système

Un cas d'utilisation spécifie une séquence d'action selon le point de vue d'une catégorie d'utilisateurs. L'étude des besoins a révélé la présence d'un acteur principale du système : L'User Agent SIP. Le diagramme de cas d'utilisation général de notre système domotique est schématisé dans la figure 4.2.

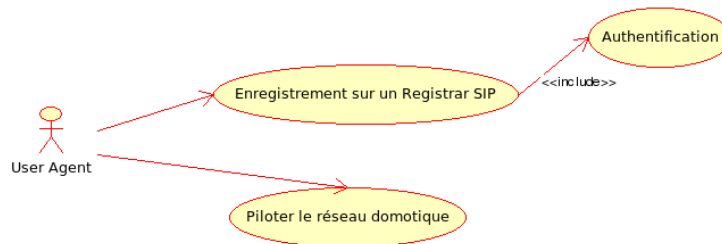


FIG. 4.2 – Cas d'utilisation général de notre système domotique

#### Cas d'utilisation «Piloter le réseau domotique»

Le fonctionnement de pilotage du réseau domotique peut être décrit par le diagramme de cas d'utilisation de la figure 4.3. Ces différentes fonctionnalités sont assurées à travers des passerelles SIP/réseau de capteurs. En effet, après son enregistrement auprès du Registrar SIP, l'Agent SIP envoie sur le réseau des messages de découverte vers les passerelles qui permettent de découvrir les dispositifs disponibles sur le réseau ainsi que leurs services.

Une fois que l'utilisateur a localisé un dispositif et ses services grâce à la phase

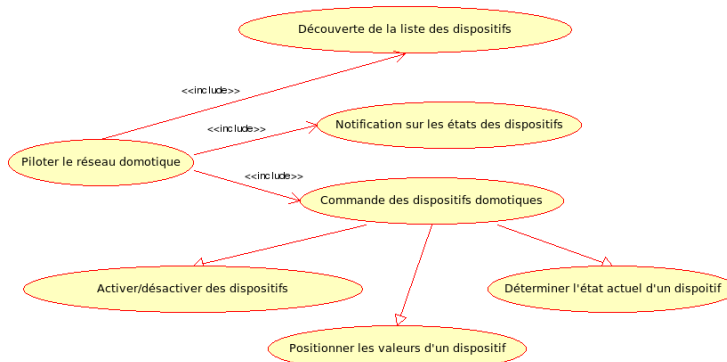


FIG. 4.3 – Cas d’utilisation «Piloter le réseau domotique»

de découverte et les informations envoyé par les passerelles, il peut procéder à la phase de souscription à un service qui lui permettra de recevoir périodiquement des notifications auprès des dispositifs domestique. L'utilisateur peut en plus envoyer des commande et éventuellement des paramètres à un service qui tentera d'effectuer la tâche due. Ces commandes peuvent être de différents types :

- Détermination de l'état actuel d'un dispositif domotique.
- Positionnement de la valeur d'un actionneur.
- Activation ou désactivation d'un dispositif domotique.

#### 4.4.3 Représentation des diagrammes de séquences

Les diagrammes de séquence permettent de décrire les interactions entre les éléments de la plateforme domestique pour chaque cas d'utilisation. Dans notre cas, ces diagrammes sont liés aux diagrammes de cas d'utilisation représentés auparavant.

##### Enregistrement de l'utilisateur distant

Quand un utilisateur veut commander son réseau domotique à distance, il doit s'enregistrer auprès du serveur d'enregistrement SIP disponible dans le réseau. Les paramètres d'authentification et les services demandés envoyés par l'agent utilisateur SIP seront vérifiées dans une base de donnée liée au serveur Registrar. Une fois la vérification effectué, un message SIP 200 OK est envoyé à l'utilisateur pour lui indiquer que l'accès est autorisé en plus d'un message indiquent la liste des passerelles sur lesquels l'utilisateur peut se connecter. La figure 4.4 donne le diagramme de séquence de ce cas d'utilisation.

##### Découverte des dispositifs existants dans le réseau

L'utilisateur peut découvrir les dispositifs disponibles sur le réseau domotique. En effet, après l'ouverture d'une session SIP avec la passerelle SIP/réseau

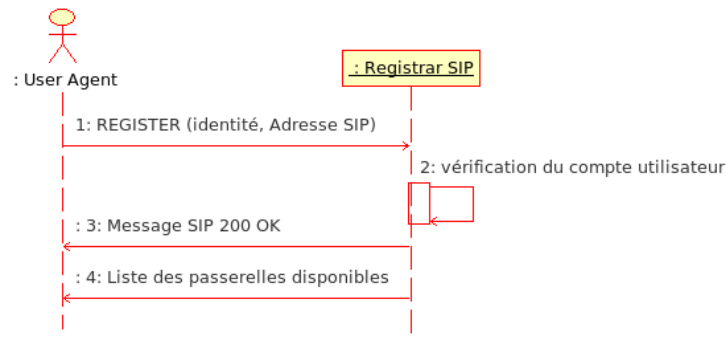


FIG. 4.4 – Diagramme de séquence «Enregistrement de l'utilisateur distant»

de capteur à travers le proxy SIP, la passerelle va consulter la base de donnée domotique pour retourner la liste des dispositifs disponibles à l'utilisateur. Cette opération est décrite par le diagramme de la figure 4.5

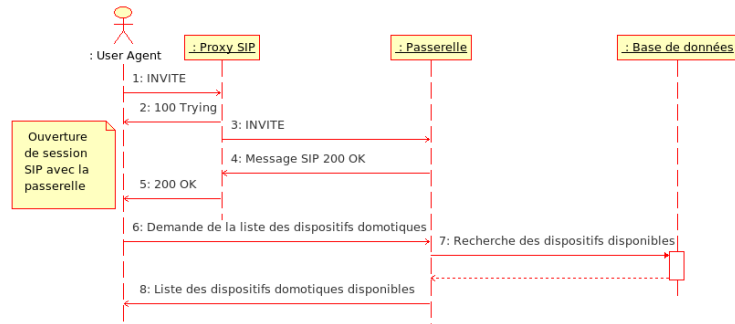


FIG. 4.5 – Diagramme de séquence «Découverte des dispositifs domotiques»

### Notification sur les états des dispositifs

Le client interroge tout d'abord la passerelle au moyen d'une requête SIP SUBSCRIBE. Si celle-ci est acceptée, le serveur en retourne le message "200 OK" et une communication est établie pendant une durée déterminée par le champ "EXPIRE" contenu dans le "Subscribe".

Une requête de notification est émise par un système auquel on demande des informations à travers une souscription. Ainsi, la passerelle va envoyer des requêtes de notifications comportant les informations à une certaine fréquence pendant la durée annoncée par le client. Cette opération est décrite par le diagramme de la figure 4.6

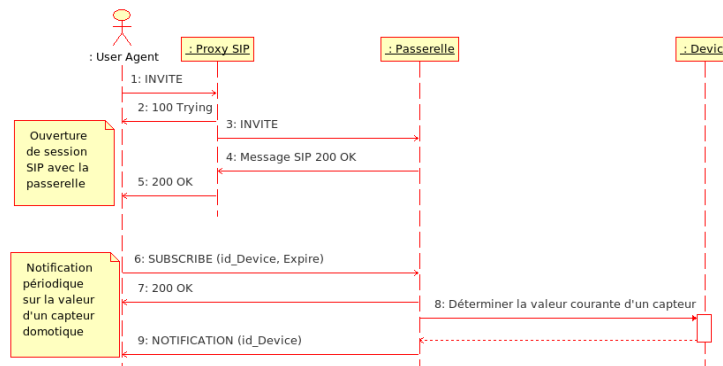


FIG. 4.6 – Diagramme de séquence «Notification sur les états des dispositifs»

### Déterminer l'état courante d'un dispositif

L'utilisateur peut savoir l'état courante d'un dispositif disponible sur le réseau domotique. Il choisit le dispositif à partir de son identificateur (id\_Device) et ensuite il valide son choix. La passerelle va consulter la base pour retourner l'état courante de l'équipement. La figure 4.7 décrit cette opération en détail.

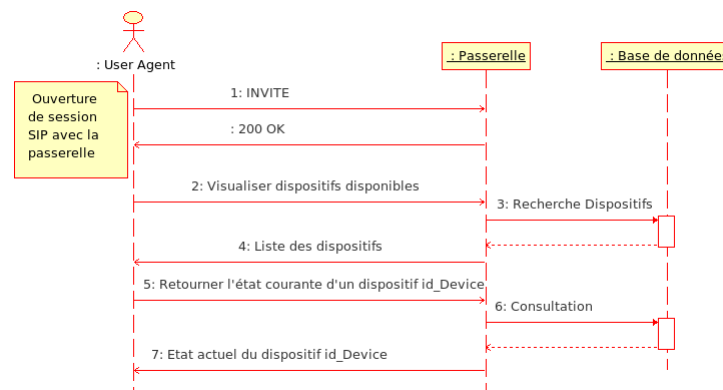


FIG. 4.7 – Diagramme de séquence «Déterminer l'état courante d'un dispositif»

### Positionner la valeur d'un dispositif

Après l'ouverture d'une session SIP avec la passerelle, l'utilisateur peut positionner les valeurs des dispositifs qu'il a la possibilité de choisir. La figure 4.8 décrit cette opération.

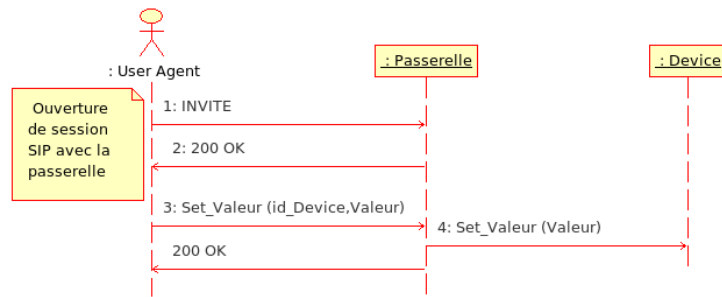


FIG. 4.8 – Diagramme de séquence «Positionner la valeur d'un dispositif»

### Activation/désactivation d'un dispositif

Pour activer un équipement dans notre plateforme domestique, l'utilisateur peut envoyer une commande ON pour un dispositif identifié par `id_Device`. Pour la désactivation, il suffit d'envoyer une commande OFF. Le changement de l'état d'un équipement doit être indiqué au niveau de la base de données de notre plateforme. La figure 4.9 décrit en détail cette opération.

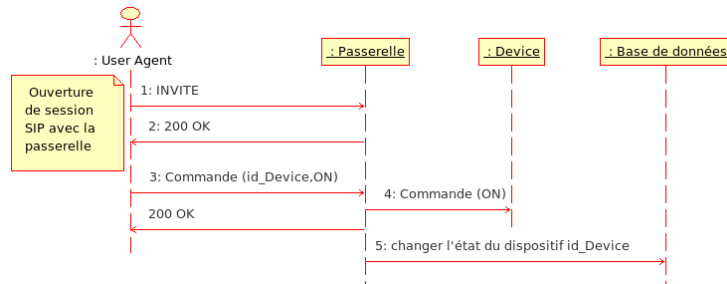


FIG. 4.9 – Diagramme de séquence «Activation/désactivation d'un dispositif»

## 4.5 Conclusion

Tout au long de ce chapitre, nous avons exposé les différents besoins auxquels doit répondre l'application à réaliser ainsi que la solution que nous allons adopter pour la satisfaction de ces exigences stipulées. Dans le chapitre suivant nous allons aborder l'implémentation de cette solution proposée tout en évoquant l'ensemble des choix techniques et logiciels qui nous ont aidés dans la réalisation de ce projet.

# Chapitre 5

## Réalisation

### 5.1 Introduction

Ce chapitre décrit la réalisation et la mise en œuvre du projet. L'organisation de ce chapitre est la suivante : dans une première partie nous présentons l'environnement matériel et logiciel utilisé, dans une deuxième partie nous détaillons les phases d'implémentation du projet en présentant les stratégies de tests adoptés pour valider l'application, la troisième partie présente les scénarios d'utilisations de la plateforme domotique.

### 5.2 Environnement matériel et logiciel

Pour développer l'application logicielle nécessaire à la mise en œuvre de la plateforme domotique, nous avons mis en œuvre différents outils afin de mieux matérialiser cette architecture.

La plateforme de développement est très proche de l'architecture du système. Elle est constituée d'un PC fonctionnant sous Linux (distribution FEDORA) relié par un HUB à des cartes cibles à base d'un processeur ARM9 développé par la société EUKREA. Le développement est effectué en premier lieu sur le PC afin de tester les programmes et d'optimiser au mieux leur empreinte mémoire, pour pouvoir les charger par la suite dans le système embarqué dans les meilleures conditions. Le rôle d'une cartes ARM sera tout d'abord d'interfacier les capteurs de température iButton et le réseau Ethernet sur la base du protocole SIP. Ensuite, elle permettra de router toutes les connections suivant les requêtes des agents SIP. La figure 5.1 présente la matérialisation de l'architecture HomeSIP.

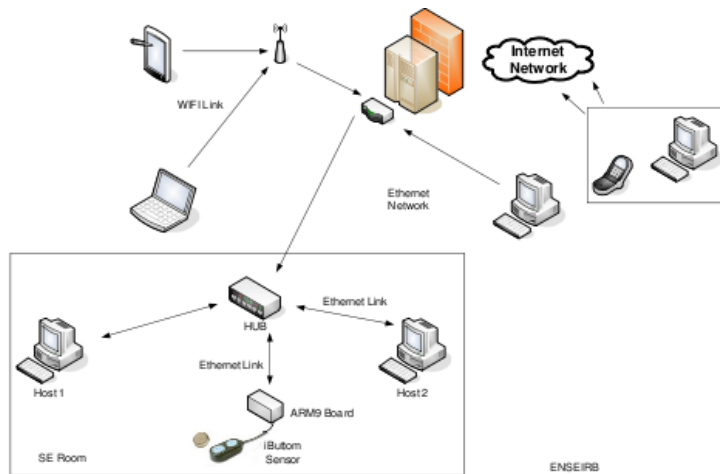


FIG. 5.1 – Matérialisation de l’architecture HomeSIP

### 5.2.1 Environnement matériel

La plateforme matérielle de notre projet est composée de :

#### Carte ARM9 d’Eukréa

Cette carte est équipée d’un système d’exploitation Linux (Distribution Linux From Scratch [22]) avec le support des périphériques du processeur et les principaux protocoles réseaux du marché. Elle possède de nombreuses E/S pour pouvoir connecter différents capteurs et actionneurs tel que les liaisons série et les bus USB. La carte possède aussi une interface réseau pour pouvoir remplir son rôle de passerelle. La figure 5.2 présente une carte ARM.

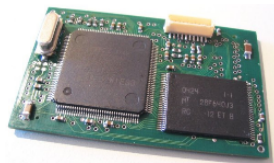


FIG. 5.2 – Microcontrôleur ARM9

#### Capteur iButton

Ces capteurs de Dallas Semiconductor permettent une mesure et un enregistrement facile de la température ou de l’humidité. Ces capteurs sont reliés au microcontrôleur à travers des bus 1-Wire. Le 1-Wire est défini comme un bus

de communication en 1 fil (plus la masse). Il a été développé par Dallas Semiconductor pour fournir une méthode de connection simple avec alimentation intégrée pour des capteurs.

Chaque composant iButton possède une clé unique 64 bits qui l'identifie. Ce numéro est également gravé sur le boîtier afin de les reconnaître de l'extérieur. Les capteurs 1-Wire de DALLAS transmettent leurs données numériques sur le bus. De plus, un mécanisme de CRC (Cyclic Redundancy Check) est prévu pour la sécurisation de la transmission des identifiants.

Il existe une multitude de iButtons remplissant différentes fonctions comme la mesure de la température ou de l'humidité, l'identification et l'enregistrement des mesures. Pour ce projet nous avons utilisé des capteurs de température iButtons DS1920. Pour la connexion avec sur le port série RS232 de la carte ARM, nous avons utilisé l'adaptateur DS9097U. La figure 5.3 présente un capteur iButton DS1920 avec adaptateur DS9097U.



FIG. 5.3 – Capteur iButton DS1920 avec adaptateur DS9097U

Dans notre projet, et pour mettre en oeuvre la technologie ZigBee dans la plateforme domotique, nous avons utilisé les modules XBee de la société MaxStream [23]. Ces modules sont certifiés ZigBee par l'Alliance ZigBee. Tirant parti de l'attrait du faible coût et de la faible consommation d'énergie du standard ZigBee, les modules XBee de MaxStream possèdent des fonctionnalités supplémentaires : différentes options de puissance de sortie, trois options d'antennes et des outils de configuration avancés.

La gamme de produits XBee est disponible pour de nombreuses applications industrielles et commerciales, parmi lesquelles la télé-détection, l'automatisation et le contrôle des équipements. Les modules XBee utilisés permettent d'émuler de manière transparente une liaison série et offrent aussi bon nombre d'I/O. Le responsable de la gestion du réseau XBee est le noeud ARM, qui prend les commandes sous format SOAP et peut exécuter les requêtes des agents SIP. La figure 5.4 présente un module XBee.



FIG. 5.4 – Module XBee

### Module CM11

Il existe plusieurs interfaces permettant de contrôler les modules X10 à l'aide d'un ordinateur. Nous avons utilisé dans ce projet des modules CM11. Ces modules peuvent s'interfacer sur les ports série ou USB de la passerelle SIP. Ces modules de commande assurent l'interface entre la passerelle SIP (carte ARM) et le réseau domotique. La figure 5.5 présente un module CM11.



FIG. 5.5 – Module CM11

### Module LM12

Ce module permet de commander une lampe en faisant varier la puissance. À l'intérieur, il utilise un triac avec détection du passage par 0 pour caler la commutation [24]. Ce module a été utilisé pour allumer des lampes dans notre infrastructure domotique. La figure 5.6 présente un module LM12.



FIG. 5.6 – Module LM11

## 5.2.2 Environnement logiciel

Dans cette section, nous présentons les différents outils de développement ainsi que les bibliothèques utilisés pour la réalisation de notre projet.

### Le cross compilateur arm-gcc

Comme pour le développement des logiciels ordinaires, pour développer des applications embarquées, nous avons besoin d'un compilateur. Dans notre cas il nous faut installer une chaîne de compilation croisée ou cross compilateur, en anglais (Cross Platform Development Tools). Le cross compilateur est un outil qui permet de compiler des programmes de manière à les faire fonctionner sur une plateforme cible dotée d'un processeur différent de celui de l'hôte. Pour notre part, nous avons décidé d'utiliser le cross compilateur arm-linux-gcc qui est le mieux adapté pour des architectures ARM.

### **Bibliothèque oSip et eXosip**

Il existe différentes piles SIP libres que l'on peut utiliser sachant qu'il convient de privilégier celles qui sont développées en langage C ou C++.

Les différents critères de choix sont : la portabilité, le respect des RFC, les couches de transport compatibles, la taille et la licence. La pile oSip est portable sur les systèmes Linux, VxWorks et Win32, elle respecte les principales RFC, elle est compatible avec les couches de transport TCP, UDP et TLS (Transport Layer Security : anciennement SSL), elle a une faible empreinte mémoire et elle possède une licence LGPL.

Le choix s'est donc porté sur la pile GNU oSIP [25] et son extension eXosip [26] développées par Aymeric Moizard. La pile oSIP étant écrite en langage C, elle est fortement portable et à faible empreinte mémoire, ce qui correspond à notre environnement embarqué. Pour l'écriture des applications SIP sur la passerelle SIP/réseau de capteur, nous avons utilisé la bibliothèque eXosip qui est une API basée sur oSIP.

### **Bibliothèque ulinuxgnu300**

Pour la mise en oeuvre des capteurs de température iButton DS1920, nous avons utilisé la bibliothèque ulinuxgnu300 [27]. Dallas Semiconductor fournit cette bibliothèque libre de fonctions C pour piloter ces capteurs à partir d'un port série moyennant l'usage d'un adaptateur bus 1-Wire/liaison série (DS9097U-S09)

### **Bibliothèque Expat**

Pour l'interprétation des corps des messages SIP qui sont structurés suivant la norme SOAP, nous avons utilisé la bibliothèque Expat [28]. Elle est multiplateforme et propose une API simple à base de callback. C'est à dire que quand un tag est trouvé, elle parse le contenu de ce tag et appelle une fonction callback. Il existe d'autres bibliothèques pour traiter des documents XML tel que LibXML2, mais LibXML2 est de grande taille, ce qui ne correspond pas aux contraintes des systèmes embarqués.

### **Logiciel Heyu**

Heyu est un logiciel permettant d'utiliser le module CM11 sous Linux. Simple d'utilisation, Heyu ne possède pas d'interface intégrée permettant de l'utiliser en mode graphique. Son utilisation en console nous a permis de l'inclure dans les scripts que nous avons développés [4].

## 5.3 Implémentation

### 5.3.1 Développement de la passerelle SIP

Nous allons décrire dans cette partie les différentes étapes de développement de la passerelle SIP :

- Portage d'une pile SIP pour la passerelle : Dans cette étape, nous avons effectué la compilation croisée des bibliothèques oSip et eXosip avec le compilateur croisé GNU C/C++ gcc pour processeur ARM fourni par Eukréa. Ensuite, nous avons installé ces bibliothèques sur la cible ARM en utilisant le protocole de transfert de fichiers TFTP.
- Compilation croisée et portage de la bibliothèque ulinuxgnu300 sur la carte ARM pour le contrôle des capteurs iButton.
- Compilation croisée et portage de la bibliothèque Expat pour le traitement des commandes SOAP envoyées à travers des messages SIP.
- Compilation croisée et portage du logiciel Heyu pour la commande des dispositifs X10.
- Après le déploiement des différents bibliothèques, nous sommes passé au développement des programmes qui prennent en charge les requêtes SIP, les exécutent sur les dispositifs domotiques et retournent les réponses SIP correspondantes.

### 5.3.2 Développement des modules ZigBee

Les phases d'implémentation des modules XBee sont :

- Programmation des microcontrôleurs PIC (Peripheral Interface Controller) afin de pouvoir assurer la communication entre les capteurs de température DS1620 et les modules émetteurs et récepteurs XBee.
- Développement d'une librairie permettant la communication entre les modules XBee par liaison série. Cette librairie contient des fonctions assurant l'écriture vers le composant et la lecture de la température envoyée par le capteur. Elle permet ainsi de configurer le module, lui demander d'effectuer une conversion ou d'envoyer la valeur de la température.
- Pour finir, nous avons fait de la compilation croisée du programme développé et nous l'avons intégré sur la carte ARM.

## 5.4 Validation de l'application

Le test est une activité indispensable pour l'obtention d'applications de qualité sujettes à des évolutions fonctionnelles régulières. C'est l'une des phases les plus importantes du développement d'un logiciel, car des bugs il y en a toujours se qu'il faut c'est que les bugs ne soit pas bloquants ou trop pénalisants pour l'utilisateur.

La démarche de test d'un projet est fortement liée à son importance et au nombre d'utilisateurs concernés. Dans notre cas, nous avons choisi d'effectuer le test unitaire.

Le test unitaire est un processus dans lequel les composants du logiciel sont examinés dans un environnement autonome (sans les autres unités). Ce test peut être réalisé avant d'examiner le logiciel dans son ensemble. Le principal avantage des tests unitaires est de nous rapprocher au maximum des erreurs sur la partie examinée du code. De ce fait, il est plus facile de localiser l'erreur (qui peut seulement être dans l'unité testée).

Nous avons adopté cette démarche qui nous a permis de déceler les erreurs rapidement en exécutant chaque méthode dès qu'elle est écrite. Ces erreurs sont donc trouvées à un stade où leur détection et leur correction sont encore faciles. Une fois une méthode écrite, nous la compilons et l'exécutons. Nous corrigeons éventuellement le code jusqu'à ce que toutes les erreurs soient corrigées, puis nous passons à la fonctionnalité suivante en adoptant la même démarche. La figure 5.7 présente la stratégie de test adoptée.

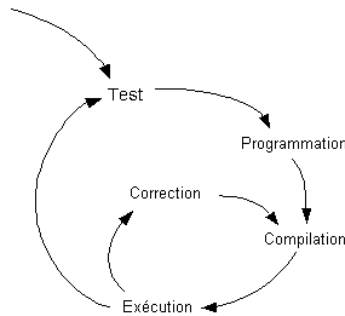


FIG. 5.7 – stratégie de test adoptée

## 5.5 Scénarios d'utilisations de la plateforme domotique

Dans cette partie, nous détaillons les différents scénarios qu'on peut l'exécuter sur la plateforme en utilisant l'outil «Wireshark» pour illustrer les traces des échanges de messages SIP entre la passerelle SIP d'adresse IP 194.199.1.200 et d'adresse SIP <sip :root@arm02.bordeaux.inria.fr> et le client SIP d'adresse IP 194.199.1.201 et d'adresse <sip :homesip@linux01.bordeaux.inria.fr>.

### 5.5.1 Découverte des listes des capteurs

Pour découvrir le liste des capteurs disponibles dans le réseau domotique, le client SIP peut envoyer une message SIP dont le corps contient la commande «getListeCapteur» selon la structure SOAP comme indiqué dans la figure 5.8.

```

MESSAGE sip:root@arm02.bordeaux.inria.fr SIP/2.0
To: sip:root@arm.bordeaux.inria.fr
From: sip:homesip@bordeaux.inria.fr
Call-ID:648324@194.199.1.201
CSeq: 1 MESSAGE
Via: SIP/2.0/UDP 194.199.1.201:5060
Content-Type: application/soap+xml
Content-Length: 381
<v:Envelope xmlns:v="http://www.w3.org/2001/12/soap-envelope" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<v:Header/>
<v:Body>
<n0:getListeCapteurs id="o0" c:root="1" xmlns:n0="urn://phoenix.labri.fr/" />
</v:Body>
</v:Envelope>

```

FIG. 5.8 – Structure SOAP de la commande «getListeCapteur»

Wireshark fournit la trace de la requête SIP sur la figure 5.9 :

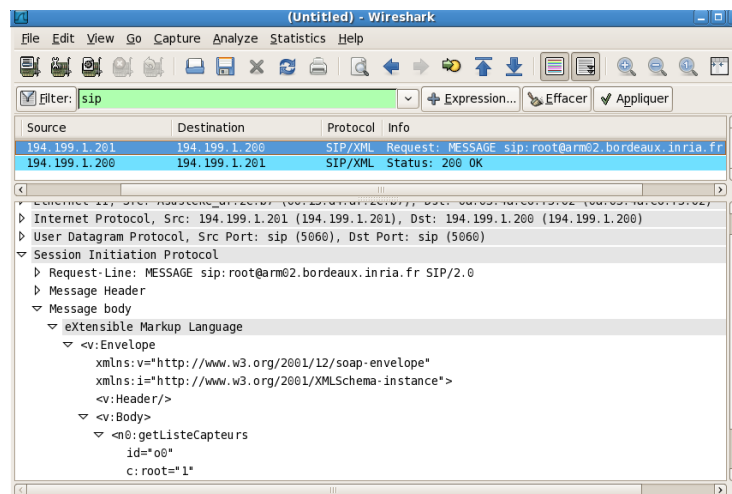


FIG. 5.9 – Trace de la commande «getListeCapteur»

La réponse de la passerelle SIP indique les capteurs disponibles, leurs types et sur quels ports ils sont connectés à la carte ARM. Dans notre cas, seul le capteur d'identificateur '0' est connecté à la passerelle SIP à travers le port série /dev/ttyS2 comme c'est indiqué par la figure 5.10.

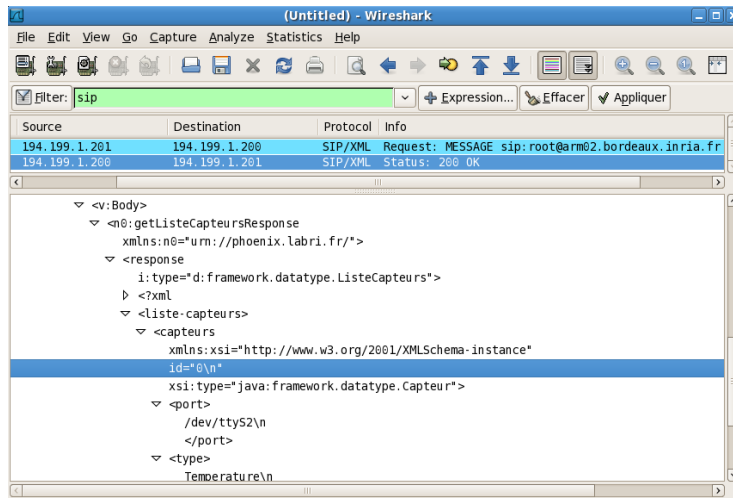


FIG. 5.10 – Réponse SIP à la commande «getListeCapteur»

## 5.5.2 Déterminer la valeur d'un capteur

Pour retourner la valeur courante d'un capteur, le client SIP envoie une requête SIP avec dans le corps SOAP du message SIP la commande `getValeurCapteur`. Dans le cas d'un capteur de température, le client SIP doit indiquer l'identificateur du capteur ainsi que l'unité de retour du résultat (Celsius ou Fahrenheit) dans le message SIP comme le montre la trace de la figure 5.11.

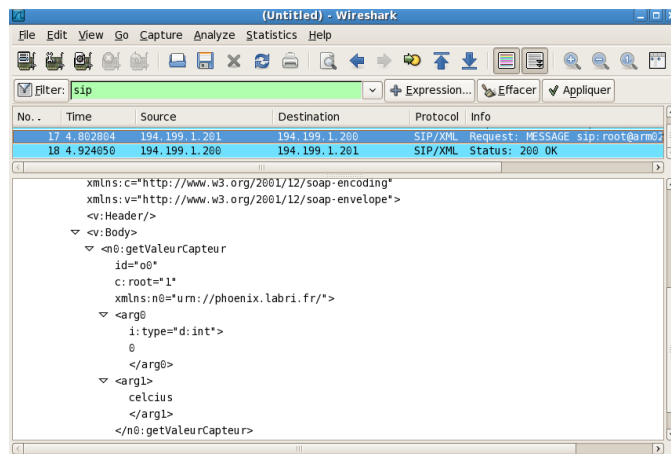


FIG. 5.11 – Trace de la commande «getValeurCapteur»



La passerelle SIP retourne les caractéristiques des dispositifs X10 disponibles sur le réseau ; les informations retournées indiquent le code site et le code dispositif de l'équipement comme le montre la figure 5.14.

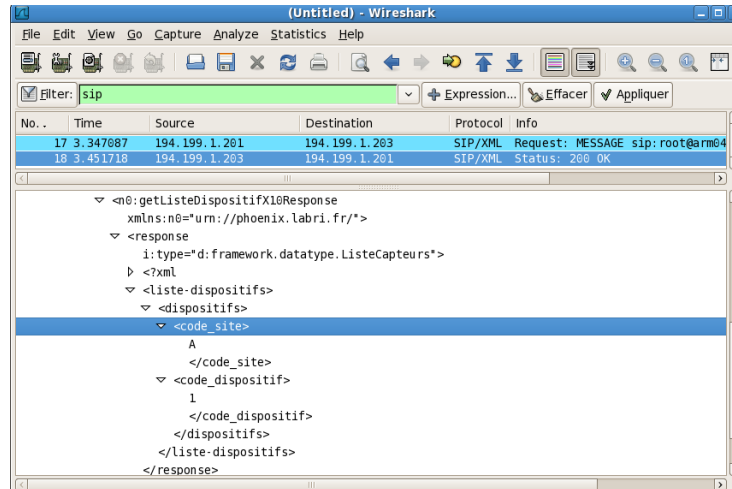


FIG. 5.14 – Réponse SIP à la commande «getListeDispositifX10»

### 5.5.4 Commande des dispositifs X10

Pour commander un équipement X10 à travers l'infrastructure SIP, le client doit indiquer le nom de la commande, le code site et le code dispositif de l'équipement X10. Le client SIP peut envoyer ensuite sa commande selon la structure SOAP de la figure 5.15.

```

<v:Envelope xmlns:v="http://www.w3.org/2001/12/soap-envelope" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <v:Header/>
  <v:Body>
    <n0:CommandeX10 id="o0" c:root="1" xmlns:n0="urn://phoenix.labri.fr/">
      <commande>on</commande>
      <code_site> A </code_site>
      <code_dispositif>1</code_dispositif>
    </n0:CommandeX10>
  </v:Body>
</v:Envelope>
  
```

FIG. 5.15 – Structure SOAP d'une commande X10

Dans notre plateforme, nous avons utilisé des modules LM12 paramétrés en A1. Ainsi la commande «on A1» permet d'allumer le module. Pour éteindre le module, nous pouvons exécuter la commande «off A1».

## 5.6 Conclusion

Nous venons ainsi de clôturer ce chapitre. En résumé, ce chapitre a présenté essentiellement les solutions adoptées lors de la réalisation de projet et la

justification de leurs choix. Nous avons terminé par l'illustration des scénarios d'utilisations de la plateforme domotique à travers les traces des messages SIP échangés entre les User Agent SIP et les passerelles SIP.

# Conclusion générale

Avec le grand progrès des technologies de la communication, de l'électronique et de l'informatique qui sont réunies pour former un seul domaine : la domotique, l'utilisateur est devenu capable de communiquer avec les différents équipements domestiques.

Dans ce cadre, nous avons essayé de développer un système domotique, qui permet aux utilisateurs de piloter et de surveiller les dispositifs domestiques localement ou à distance à travers une infrastructure SIP.

Ce protocole fournit des fonctionnalités qui permettent de découvrir et d'utiliser les différents services dans un réseau domotique d'une manière automatique. En effet, nous avons détaillé l'utilisation de SIP en étudiant les étapes de son fonctionnement (adressage, description et contrôle, etc..) et les différents mécanismes qui permettent la coordination entre les divers dispositifs câblés (sondes, appareils photos,...) et sans fil et qui prennent en considération l'hétérogénéité des périphériques utilisées (capteurs, dispositifs X10, caméras,...).

L'apport de notre projet se résume surtout dans l'intégration de différentes technologies domotiques existantes tels que X10 et ZigBee à travers la mise en oeuvre du protocole SIP dans un environnement embarqué. Notre solution s'inscrit dans la convergence de l'informatique et de l'électronique et met en pratique cette notion. Dans ce cadre, le projet sera soumis pour la conférence ICC 2009 qu'organise l'IEEE à *Dresden en Allemagne* en juin 2009 [29].

Finalement, et pour clore ce mémoire, nous espérons que les objectifs fixés au départ ont été en grande partie réalisés et que les résultats obtenus trouveront un bon écho.

# Bibliographie

- [1] Hager Group, <http://la.domotique.online.fr/index.htm>, 2001.
- [2] Mark Weiser, "The computer for the 21st century, ACM SIGMOBILE Mobile Computing and Communications Review," <http://doi.acm.org/>, Vol. 3, no. 3, 1999, pages 3–11
- [3] [www.guideinformatique.com/definition-appliance-453.htm](http://www.guideinformatique.com/definition-appliance-453.htm)
- [4] GNU/Linux Magazine, Numéro 103 - Mars 2008 , pages 22-31
- [5] [www.electronique.biz](http://www.electronique.biz)
- [6] [www.wikipedia.org/wiki/LonWorks](http://www.wikipedia.org/wiki/LonWorks)
- [7] CEBus Industry Council, [www.cebuse.org](http://www.cebuse.org), 2001
- [8] <http://fr.wikipedia.org/wiki/HomeRF>
- [9] <http://www.zigbee.org/>
- [10] Technical White Paper : Jini Architectural Overview, 2000. <http://www.sun.com/jini>.
- [11] Guttman, E., Perkins, C., Veizaldes, J., and Day, M., "Service Location Protocol, Version 2," IETF, RFC 2608, June 1999. <http://www.rfc-editor.org/rfc/rfc2608.txt>
- [12] Microsoft Corporation. Understanding Universal Plug and Play : a White Paper, June 2000. <http://www.upnp.org/ressources/whitepapers.asp>.
- [13] UPnP Forum. UPnP Device Architecture 1.0. UPnP Forum, May 2003.
- [14] <http://www.idate.fr>
- [15] [www.eukrea.com](http://www.eukrea.com)
- [16] [www.maxim-ic.com/products/ibutton/](http://www.maxim-ic.com/products/ibutton/)
- [17] <http://rfceditor.org/>
- [18] IETF RFC 3261, "All-IP Core Network Multimedia Domain - IP Multimedia Call Model", Stage2, 07-2003
- [19] International Softswitch Consortium, Reference Architecture, June 2002. [www.softswitch.org](http://www.softswitch.org)
- [20] Pascal Roque, «les cahiers de programmeur UML», Edition Eyrolles, 2001.
- [21] Pierre-Alain Muller, «Modélisation Objet avec UML», Edition Eyrolles, 1998

- [22] <http://lfs.traduc.org/>
- [23] <http://www.maxstream.net/>
- [24] <http://fribotte.free.fr/bdtech/Domotique/Domotique.html>
- [25] <http://www.gnu.org/software/osip/>
- [26] <http://savannah.nongnu.org/projects/exosip/>
- [27] [http://files.dalsemi.com/auto\\_id/public/ulinuxgnu300.zip](http://files.dalsemi.com/auto_id/public/ulinuxgnu300.zip)
- [28] <http://www.jclark.com/xml/expatfaq.html>
- [29] <http://www.ieee-icc.org/2009/>