

Rémi GAREL

remi.garel@gmail.com

Université Jean-Monnet

Ecole des Mines de Saint-Etienne

Rapport de stage – Posintra

Master Informatique Web Intelligence, Saint-étienne



Période : Février 2009 - Juin 2009

Tuteur en entreprise : Kary Främling

Tuteur à la faculté : Olivier Boissier



SÄHKÖISEN TALOTEKNIIKAN OSAAMIS- JA KEHITTÄMISKESKUS

Posintra Oy
Kehitysyhtiö
Kipinätie 1
06150 Porvoo
+358 10 8367 700
info@posintra.fi

Remerciements

Pour commencer ce rapport, il est indispensable de remercier certaines personnes qui ont rendu ce stage possible et agréable. Tout d'abord je tiens à remercier Kary Främling pour m'avoir accueilli au sein de son projet, à Posintra. Je voudrais également remercier Jan Nyman pour ses réponses à mes questions et sa sympathie. Kariuki M'Botha pour sa bonne humeur et ses leçons d'anglais.

Fredrick von Schoultz pour m'avoir accepté dans sa société et Leena Tuomi pour avoir été une directrice très attentionnée.

J'aimerais avoir une attention particulière pour Jemina Tuominen, Ylva Hanula et Jenni Lähtevänoja qui m'ont été précieuses dans la recherche d'appartement et les papiers administratifs. Ainsi que l'ensemble des employés de l'entreprise Posintra pour leur gentillesse.

Pour finir, je souhaite remercier tout particulièrement certains professeurs de mon université et de l'école des mines. François Jacquenet pour avoir arrangé la fin de l'emploi du temps afin de me permettre de terminer mon stage. Colin De La Higuera qui a su passer beaucoup de temps pour me chercher un stage à l'étranger, et enfin et surtout, Olivier Boissier qui a également pris de son temps pour contacter Kary et qui a été très attentif tout au long de mon stage.

Résumé

Le raison de ce rapport est de mettre en place un système de communication inter-périphériques dans le but de faire des économies d'énergie. Pour ce faire, je mettrai en avant un ensemble de scénarii ayant pour objet la gestion de l'électricité et de l'eau et présenterai la norme oBIX ainsi que la manière dont je l'ai intégré à DIALOG de manière à avoir un historique qui peut s'avérer utile pour les analyses de données.

Mots clés : Informatique diffuse, oBIX, économies d'énergie

Abstract

The purpose of this report is to create a system of communication between several devices. The goal is to save energy. To create this communication, I will expose some scenarii which are related to electricity and water management. I will continue with a presentation about the oBIX norm and how I integrated it to DIALOG in order to have an history which can be usefull to analyse datas.

Keys Words : Ubiquitous computing, oBIX, energy preservation

Table des matières

Remerciements	3
Abbréviations	9
Introduction	11
I La communication inter-périphériques dans le cadre d'une amélioration des conditions de vie et la réduction de la consommation d'énergie	13
1 Concepts	15
1.1 Informatique diffuse	15
1.2 Internet Of Things	16
2 Etat de l'art	17
2.1 Informatique diffuse	17
2.2 La maison intelligente	17
2.3 La communication	18
2.3.1 Les protocoles	19
3 Apport d'une solution	21
3.1 Les capteurs	21
3.1.1 Capteurs de luminosités	21
3.1.2 Capteurs thermiques	21
3.1.3 RFID	22
3.2 Les améliorations de consommation d'énergie	22
3.2.1 Consommation électrique des équipements et de l'éclairage	22
3.2.2 Consommation d'eau	23
3.3 La communication entre les différents capteurs, périphériques et le(s) serveur(s)	23
4 Ouverture	25
4.1 L'informatique diffuse en dehors de la maison	25
4.2 La sécurité et le respect de la vie privée	25
4.3 Apparition de nouveaux périphériques dans une zone	26
4.4 Problème de localisation des acteurs	26
4.5 Action - Réaction	26

II	Mon travail	29
5	Mise en place d'un serveur oBIX	31
5.1	Background	31
5.2	Qu'est ce que oBIX ?	31
5.3	Les concurrents de oBIX	32
5.4	Le serveur oBIX	33
6	Intégration de oBIX a DIALOG	35
6.1	Qu'est ce que DIALOG ?	35
6.2	oBIX dans DIALOG	35
6.3	La démonstration en place à Posintra	37
	Conclusion et perspectives	39
A	Présentation de l'entreprise	41
A.1	Posintra Oy	41
A.2	STOK	41
A.3	Le personnel	42
A.4	Les partenaires	42
A.4.1	Enervent	42
A.4.2	Nokia	42
A.4.3	Ensto	43
A.4.4	Oras	43
A	Extra	45
A.1	Le séminaire UBI Summit 2009	45
A.1.1	Première intervention : Hanna Marttinen-Deakins	45
A.1.2	Deuxième intervention : Nicholas R. S. Evans	45
A.1.3	Troisième intervention : Ryan Genz et Francesca Rosella	46
A.2	La journée de Posintra	46
A.2.1	La visite de l'université de Helsingki	46
A.2.2	La "Design Factory"	46
A.2.3	Le RFID Labs	47

Abbreviations

DIALOG Distributed Information Architectures for collaborative logistics

HUT Helsinki University of Technology

HTTP HyperText Markup Language

HVAC Heating, ventilating and air conditioning

IOT Internet Of Things

OASIS Organisation for the Advancement of Structured Information Standard

oBIX Open Building Information eXchange

oFMS Open Facility Management Server

PMI Promise Message Identification

REST Representational State Transfer

RMI Remote Method Invocation

RFID Radio Frequency IDentification

TKK Teknillinen korkeakoulu (équivalent à HUT)

XML Extensible Markup Language

Introduction

Dans le cadre de mon master Informatique Web Intelligence, il m'était demandé d'effectuer un stage de fin d'étude permettant de mettre en pratique les différents acquis universitaires.

J'ai décidé de faire ce stage à l'étranger dans le but d'améliorer mon anglais qui est une langue indispensable dans le domaine de l'informatique. C'est ainsi que je me suis retrouvé dans la ville de Porvoo en Finlande au sein de la société Posintra grâce à la relation existante entre Kary Främbling et Olivier Boissier.

Hormis la possibilité d'améliorer mon anglais, j'ai choisi ce stage pour le sujet, en effet, j'avais également eu la possibilité d'effectuer un stage à Londres mais le sujet n'était pas des plus intéressant, à l'inverse du sujet proposé par Kary Främbling, sujet ayant un rapport avec l'informatique diffuse, domaine que j'ai pu tout particulièrement apprécier cette année lors de mes cours à l'école des mines et notamment avec les cours de Fano Ramparany ainsi que l'exposé nous ayant permis de découvrir le projet AMIGO.

Ainsi le but de ce stage a été d'insérer la norme oBIX à DIALOG. Effectivement, un des problèmes principaux de l'informatique diffuse est la communication inter-périphériques, et, avoir un serveur respectant un maximum de normes de communication est indispensable. Au delà de l'intégration de oBIX au sein du serveur, je me suis penché sur la question de l'économie d'énergie pouvant être engendrée grâce à cette communication. J'ai ainsi mis en place différents scénarii dans lesquels je tente de montrer quelles seraient les étapes indispensables à leur développement.

Dans un premier temps, le rapport débutera par une définition d'un certains nombres de concepts indispensables à la compréhension du domaine. Elles seront suivies par une étude bibliographique du domaine qui débouchera sur les scénarii envisagés pour diminuer la consommation d'énergie. Ces scénarii m'ont conduit à envisager quelques questions ayant un rapport moins direct avec mon sujet que j'ai donc placé en ouverture.

Ensuite, nous aborderons la norme oBIX, comment les objets sont structurés et comment la communication est effectuée pour finir sur l'intégration de cette norme à DIALOG.

Première partie

La communication inter-périphériques dans le cadre d'une amélioration des conditions de vie et la réduction de la consommation d'énergie

Chapitre 1

Concepts

1.1 Informatique diffuse

L'informatique diffuse est connue sous plusieurs noms : informatique pervasive , informatique ubiquitaire, informatique omniprésente (sans compter les traductions anglaises fréquemment utilisées, notamment "Ubiquitous computing"). Derrière l'ensemble de ces termes se cache néanmoins le même concept dont voici quelques définitions collectées au long de mes recherches.

Dans un premier temps, cette vision nouvelle de l'informatique a été initié par Mark Weiser dans un document datant de 1991 intitulé "The computer for the 21st Century" [18].

Dans ce papier, la première phrase résume à elle seule le but de l'informatique diffuse : "La plus profonde des technologies est celle qui disparaît". Ainsi le but ici, est d'intégrer l'informatique à l'environnement de manière à ce que l'Homme ne se rende plus compte de son utilisation. A cette époque, Mark Weiser indiquait un certain nombre de problèmes comme le manque de mémoire, les tailles de composants, dont il savait que l'évolution des technologies permettrait de s'acquitter mais aussi d'autres problèmes dont nous reparlerons tout au long de ce rapport.

De nos jours, le principe est défini d'une manière équivalente, mais les problèmes techniques énoncés par Mark Weiser ont disparu et laissé place à des problèmes conceptuels.

Journal du net : Tendances vers l'informatisation, la connexion en réseau, la miniaturisation des dispositifs électroniques et leur intégration à n'importe quel objet du quotidien, favorisant ainsi l'accès aux informations dont on a besoin partout et à tout moment. L'informatique diffuse fait référence à l'utilisation de plus en plus répandue de processeurs minuscules communiquant spontanément les uns avec les autres et de capteurs qui, grâce à leurs dimensions très réduites, seront intégrés dans les objets de la vie quotidienne, jusqu'à devenir presque invisibles pour les utilisateurs.

[3] : L'informatique diffuse a pour but d'offrir des services aux utilisateurs humains interagissant avec leur environnement (y compris les objets et autres humains qui l'occupent). Les applications dans ce domaine doivent être capable de considérer le contexte dans lequel les utilisateurs évoluent (qu'il s'agisse de leur localisation physique, leur position sociale ou hiérarchique ou leurs tâches courantes ainsi que des informations qui y sont liées). Ces applications doivent gérer dynamiquement l'irruption dans la scène de nouveaux éléments (utilisateurs ou appareils) même inconnus et produire de l'information de contexte utile à des applications non envisagées.

1.2 Internet Of Things

L'Internet Of Thing (IOT) est une vision du monde où tous les objets présents auraient un moyen leur permettant d'être identifiés, en utilisant par exemple, des codes à barres ou des puces RFID. Cette dernière solution semble de plus en plus s'imposer, avec l'arrivée de puces de plus en plus petites grâce aux évolutions des nanotechnologies. L'arrivée de l'IP v6 est également une avancée importante pour l'Internet Of Thing, l'IP v4 nous limitant dans le nombre d'adresses IP disponibles, ce problème ne se posera désormais plus.

Dans ce domaine, on pointe principalement la question suivante : Doit-on créer des objets intelligents qui sont capables de décider tout seul de leurs agissements, ou les objets doivent-ils simplement être capable de communiquer leurs états et réagir aux messages reçus ? [10] [5]

Chapitre 2

Etat de l'art

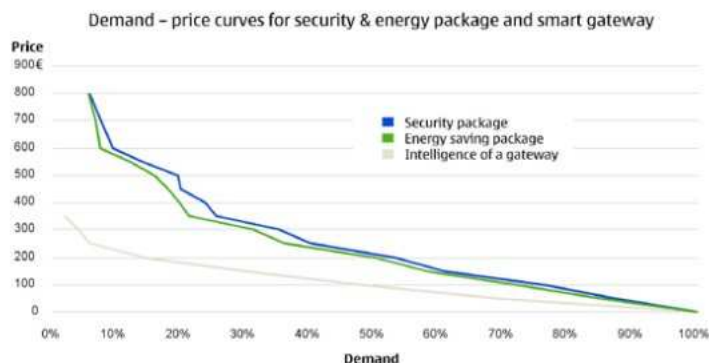
2.1 Informatique diffuse

L'apparition de la vision d'informatique diffuse, a eu lieu en 1988 dans une conférence menée par Mark Weiser, il s'en est suivi le papier cité précédemment [18], dans lequel il fait un parallèle avec les mondes virtuels, tout en affirmant que ce dernier n'aurait pas un effet bénéfique, en effet, il entrainerait un repli sur lui même de l'être humain, Mark Weiser préfère ainsi imaginer une informatique qui assiste la famille lors de la préparation du petit déjeuner par exemple. Il n'oublie naturellement pas de citer un aspect indispensable au bon fonctionnement de l'ubiquitous computing : la sécurité. Ainsi il propose simplement de crypter les échanges. Cet article sera la base de beaucoup d'autres dont [13] qui cherche à approfondir le sujet maintenant que nous sommes dans le XXIème siècle.

[14] d'une part et [1] d'autre part, sont deux articles se basant sur les travaux de Mark Weiser où on peut trouver une liste de challenges dont nous reparlerons régulièrement.

2.2 La maison intelligente

La maison intelligente est une des applications possibles de l'ubiquitous computing, c'est un des rêves de l'Homme moderne. Néanmoins, malgré les grandes avancées technologiques de l'informatique, nous sommes encore loin de ce dont rêvait nos prédécesseurs avec la création de robots qui gèreraient l'intégralité des travaux domestiques. Mais avons-nous réellement envie d'une telle évolution ? Il semblerait que non, une étude de 2008 établie par Nokia tend à montrer que les gens sont prêt à dépenser de l'argent principalement pour une sécurité accrue, ainsi qu'une baisse de consommation d'énergie, plutôt qu'une amélioration de leur confort.



Ainsi la plupart des travaux effectués dans le domaine de la recherche sont effectués principalement dans le but de réduire la consommation d'énergie et d'augmenter la sécurité des habitats. Néanmoins, malgré les grands avancements qui ont eu lieu dans le domaine de nombreuses pistes restent à explorer.

La maison intelligente est abordée de différentes manières, pour le moment de grands industriels tentent de faire figuration en installant des modèles préfabriqués où tout est contrôlable grâce à une télécommande (PDA, téléphone, écran tactile), dans les meilleurs des cas via des navigateurs Internet, ce qui permet de contrôler sa maison même lorsqu'on est absent. Ces systèmes sont très coûteux. En réalité, créer une maison intelligente est bien plus compliqué. Les gens ne souhaitent pas changer l'ensemble de leurs mobiliers et objets hi-tech, mais plutôt de les adapter et surtout, ils ne souhaitent pas être lié à un fabricant particulier. Le problème qui se pose alors, est la mise en place d'un système de communication cohérent entre différents objets n'utilisant pas nécessairement les mêmes protocoles.

Effectivement, il est fréquent de voir des fabricants utiliser différents protocoles car rien n'est standardisé. Il a donc été nécessaire de créer des Middlewares, comme l'a fait AMIGO, un grand projet européen visant à faire avancer les recherches dans ce domaine, en regroupant de grandes entreprises (entre autres : France Telecom, Philips, Microsoft, Fagor...) De ce projet est né un middleware dont on peut voir les nombreuses démonstrations sur le site officiel ¹ du projet.

([9] et [16])

Un autre grand projet européen a pris la succession de AMIGO depuis que ce dernier c'est terminé, il porte le nom de SOPRANO². [8]

2.3 La communication

Comme il a été décrit précédemment, le principal problème de l'informatique pervasive est la communication des différents composants. Dans cette partie je vais exposer une manière de regrouper différents type de messages.

oSSG Un oSSG (open Source Service Gateway) est ce qui permet au serveur de faire communiquer différents objets. Ainsi, il existe entre autres oFMS et

¹<http://www.hitech-projects.com/euprojects/amigo/>

²<http://www.soprano-ip.org/>

DIALOG que je vais détailler ci-dessous.

- oFMS

Open Facility Management Server (oFMS) est un équipement de serveur interopérable pour les systèmes automatiques des maisons ou immeubles. Il est basé sur oBIX (open Building Information eXchange). Les informations sont stockées dans un format oBIX et le serveur permet d'accéder à l'interface du Web Service pour les sous systèmes (périphériques) et pour le système de contrôle (interface utilisateur).

Au niveau de l'intégration de la plateforme, le serveur est placé entre le système de contrôle et les sous systèmes indépendants qui communiquent via des protocoles variés. Il nivelle les informations et contrôle le format oBIX.

Les caractéristiques de l'implémentation du serveur sont définies par les spécifications oBIX. La connexion est établie en utilisant des adapteurs qui traduisent différents protocoles au format oBIX. oFMS est donc un serveur autonome au lieu d'être uniquement intégré à un système de contrôle.

- DIALOG

DIALOG quant à lui reprend l'ensemble des points gérés par oFMS, hormis la compatibilité avec oBIX, et ajoute à cela la possibilité de conserver les données, chaque noeud étant relié à une base de données. DIALOG permet également de souscrire à des abonnements, ainsi, par exemple, si une voiture a un problème, le garagiste peut s'abonner à l'ordinateur de bord de la voiture, et ce dernier lui enverra les informations souhaitées lorsqu'elles arriveront.

Les formats de message

- oBIX

oBIX est une initiative de grandes industries pour définir des mécanismes XML, et web services, pour le contrôle de systèmes habités. Le but de oBIX est de définir un standard pour les protocoles de web services pour rendre possible la communication entre la partie mécanique, le système électrique et les applications.

- PMI

Les messages PMI sont en réalité un moyen d'englober les messages, ils sont utilisés actuellement par DIALOG, ils sont en train d'être examinés pour être reconnus comme norme.

2.3.1 Les protocoles

L'ensemble des protocoles de communication entre divers périphériques est assez conséquent, on remarque notamment les protocoles de types REST. REST qui signifie Representational State Transfer, est une manière de construire une application pour les systèmes distribués c'est un style d'architecture, bâti sur quelques principes simples :

- connaître l'URI doit suffire pour accéder à la ressource ;
- chaque opération est auto-suffisante : il n'y a pas d'état ;
- HTTP un protocole sans état, avec un nombre très limité d'opérations,

REST concerne l'architecture globale d'un système. Il ne définit pas la manière de réaliser dans les détails. En particulier, des services REST peuvent être réalisés en .NET, JAVA, CGI ou COBOL.

Chapitre 3

Apport d'une solution

L'ensemble des capteurs présents sur un périphérique peuvent être gérés par le périphérique lui-même ou par un serveur qui collecte les informations. Dans le cadre de oBIX le but est de modéliser l'ensemble des données qui peuvent être fournies par différents capteurs et de les fournir au serveur. Le serveur peut également s'il le souhaite invoquer des méthodes à distance. Ce moyen permet ainsi de gérer l'ensemble des capteurs existants, que ce soit les puces RFID, thermomètres ou tout autres capteurs censés ne fournir que des informations simples. En effet, le serveur collecte les données et peut ensuite les analyser ou faire agir un périphérique en fonction de ces résultats.

3.1 Les capteurs

3.1.1 Capteurs de luminosités

Il existe différents types de capteurs de luminosités

- les photodiodes : composant semi-conducteur ayant la capacité de détecter un rayonnement du domaine optique et de le transformer en signal électrique.
- les capteurs photographiques : composant électronique photosensible servant à convertir un rayonnement électromagnétique (UV, visible ou IR) en un signal électrique analogique. Deux grandes familles de capteurs sont disponibles : les CCD et les CMOS.
- les cellules photoélectriques : dispositif composé d'un capteur photosensible dont la résistance électrique varie lorsqu'il est soumis à un rayonnement lumineux.

3.1.2 Capteurs thermiques

- les thermomètres : un des capteurs que nous connaissons tous très bien.
- les pyranomètres : capteur de flux thermique utilisé pour la mesure de la quantité d'énergie solaire en lumière naturelle et est notamment utilisé en météorologie. Il permet la mesure de la puissance du rayonnement solaire total en watts par mètre carré.

3.1.3 RFID

Ce sont des dispositifs passifs, ne nécessitant aucune source d'énergie en dehors de celle fournie par les lecteurs de tags. Le tag est extrêmement discret avec sa taille réduite (quelques millimètres), et sa masse négligeable. Son coût étant devenu minime, on peut envisager de le rendre jetable, bien que la réutilisation soit une solution plus écologique.

Le marqueur se compose :

- d'une antenne ;
- d'une puce de silicium ;
- d'un substrat et/ou d'une encapsulation.

Notons aussi qu'il existe des tags "actifs" et des "semis-actifs" qui incluent une batterie permettant une communication avec des distances plus élevée.

3.2 Les améliorations de consommation d'énergie

3.2.1 Consommation électrique des équipements et de l'éclairage

Contexte Dans une maison, le nombre d'appareils électriques et de lampes est assez conséquent. L'ensemble de ces éléments constituent une bonne part de la facture d'électricité, il est donc indispensable de trouver une solution permettant de réduire les coûts. On imagine bien que l'extinction des lampes lorsqu'on navigue dans une maison n'est pas une des principales occupation des habitants. De la même manière, les appareils tel que les télévisions, les radios, et bien d'autres peuvent s'avérer très consommateur d'énergie par manque de motivation des occupants de la maison.

Objectifs L'objectif serait donc de mettre en place un système qui en fonction des déplacements effectués dans la maison ainsi que de la luminosité extérieure pourrait allumer ou non les lampes de la maison sans que les occupants n'aient à se soucier d'appuyer sur un interrupteur. Il en est de même avec les appareils électo-ménagés, mais ici naturellement le capteur lumineux n'est pas nécessaire.

Scénarii Il y a deux scénarii possibles :

- La luminosité extérieure est suffisante, dans ce cas la lumière ne s'allumera pas où que soit les différents acteurs
- La luminosité extérieure est faible, (ou on se trouve dans les cas d'appareils électo-ménagés) il faut alors allumer uniquement les lampes (respectivement, appareils) situées dans les zones d'action des habitants.

Mise en place La mise en place d'un tel dispositif n'est pas compliqué, surtout si on dispose d'un système permettant la réception de données et des appels de méthodes à distance. Un serveur oBIX permet cela. Il suffit pour chacun des équipements de créer un adaptateur :

- Dans le cas des capteurs (les capteurs lumineux et le sol sensitif), l'adaptateur devra envoyer en permanence les données collectées pour savoir dans quel état on se trouve
- Dans le cas des appareils à contrôler, il faudra créer un adaptateur écoutant la liste des changements perpétuellement afin de basculer dans l'état adéquat.

Le serveur devra connaître les zones de mise en action de chacun des appareils et à chaque entrée d'un personnage dans cette zone, les déclencher, ou les éteindre en cas de sortie de cette zone d'action

3.2.2 Consommation d'eau

Contexte De la même manière que l'électricité, l'eau constitue une grande partie des dépenses relatives aux maisons. Ici bien que le même système pourrait être mis en place, il pourrait être ennuyeux d'obliger quelqu'un à rester devant le robinet si il souhaite remplir un seau par exemple. On pourrait néanmoins appliquer le même système que l'électricité pour certains robinets de la maison. Ici, le cas le plus important de dépenses inutiles seraient plutôt les fuites d'eau. Ainsi, il pourrait être judicieux de vérifier les consommations des différents robinets et en cas de consommation étrange prévenir le plombier (ou conseiller l'utilisateur d'appeler un plombier) Un autre moyen d'éviter les fuites, peut être de couper les robinets d'eau en cas d'absence des occupants.

Objectifs Le but est donc de limiter les risques de consommation d'eau involontaire via un système de localisation de personnes pour certains robinets et d'analyse de consommation d'eau. En cas d'absence des occupants de la maison, il suffit de couper l'eau pour limiter les risques (on peut également imaginer bloquer l'arrivée d'eau lorsque les habitants dorment)

Mise en place Encore une fois la mise en place de ce dispositif reste assez aisée.

Le sol sensitif permet de savoir si des personnes sont présentes ou non dans la maison, mais également de connaître leur position dans la maison. Ainsi, il est possible de récupérer les données sous format XML et de les traiter en fonction des paramètres de chacun des robinets. Il faudrait tout de même prévoir de coupler ce système à des détecteurs de fumée et des détecteurs de monoxyde de carbone permettant de réactiver l'arrivée d'eau. (Ici pourrait se poser le problème de la priorité des messages envoyés, pour savoir si il faut plutôt ouvrir ou fermer les robinets devant la présence de deux messages différents)

3.3 La communication entre les différents capteurs, périphériques et le(s) serveur(s)

Naturellement, l'ensemble des capteurs ne parlent pas le même langage, c'est pourquoi la création d'adapteurs sera indispensable. Ces adapteurs seront comme les pilotes qu'on connaît actuellement lorsqu'on branche une imprimante sur son ordinateur par exemple. Ces capteurs devront transformer l'ensemble des données dans le format de message de la norme oBIX. Puis gérer la connexion via HTTP entre le périphérique et le serveur. Le serveur une fois en possession de l'ensemble des données pourra réagir et modifier les différents états des périphériques via la queue de changement d'oBIX qui sera scanné en permanence par les adapteurs.

Chapitre 4

Ouverture

4.1 L'informatique diffuse en dehors de la maison

L'informatique omniprésente n'est naturellement pas une technologie réservée aux maisons, ainsi les universités [19] qui contiennent un nombre important d'ordinateurs, dû à la présence de nombreux étudiants et professeurs, pourraient voir apparaître une mise en relation entre chaque machine. Par exemple lorsqu'un professeur écrit au tableau, l'étudiant reçoit directement le cours sur son ordinateur et si il a un problème, il peut pointer facilement la partie mal comprise sur son ordinateur mais le professeur la verra lui directement sur le tableau.

On peut également imaginer la mise en place d'une telle technologie dans les musées [4], ou les visiteurs pourraient par exemple se servir de leur téléphone pour collecter des informations sur les oeuvres contemplées, on peut même imaginer encore mieux avec la réalité augmentée [11] qui consiste à placer les informations directement sur les objets grâce un dispositif particulier, une caméra par exemple, qui reconnaît l'objet perçu et va chercher les informations les concernant. Ici on approche à la fois la reconnaissance de forme et le web sémantique pour la collecte d'informations.

4.2 La sécurité et le respect de la vie privée

Comme n'importe quel composante de l'informatique, l'informatique diffuse n'échappe pas à la règle de la sécurité et de la préservation des données relatives à la vie privée, ainsi on peut trouver dans [17] et [15], un ensemble d'éléments permettant de prendre conscience des risques de l'ubiquitous computing. L'exemple le plus parlant reste néanmoins le cas de la gestion du chauffage et de la ventilation. En effet, pour un tel système, il est nécessaire de savoir si les personnes qui habitent la maison sont présentes, absentes ou absentes pour une longue durée, ce dans le but de gérer au mieux la communication entre les deux systèmes (aérer lorsque le chauffage ne fonctionne pas et donc en cas d'absence prolongée). Mais on imagine bien qu'une personne mal intentionnée, si elle avait accès à ces données sensibles pourraient cambrioler la maison tout en sachant que l'ensemble de la famille ne rentrera pas avant une longue période, et donc

n'aurait aucun risque de se faire surprendre. Il existe néanmoins une solution pour palier à ce problème, un détecteur de présence qui en cas d'absence de personnes dans la maison et d'intrusion, déclenche l'alarme et envoi un texto au propriétaire de la maison (on peut même imaginer un appel direct au commissariat, ou une connexion à une webcam permettant de voir qui se trouve dans la maison).

4.3 Apparition de nouveaux périphériques dans une zone

Que ce soit dans une université, un musée, une maison ou tout autre endroit où on souhaite insérer de l'informatique ubiquitaire, il est nécessaire de détecter les nouveaux périphériques. Et ceci engendre deux nouveaux problèmes, le premier est la méconnaissance du protocole associé au périphérique. En effet, si on branche un périphérique sur un port USB "Plug and Play", on s'attend à un type de signal, dans le cas de l'informatique diffuse, les périphériques cherchant à se connecter ne suivent pas forcément un protocole unique, dans le cas des périphériques sans fils par exemple, certains peuvent essayer de se connecter par Wifi, d'autres par Bluetooth, Zigbee et bien d'autres technologies encore, on imagine donc la difficulté de cette détection, [2] tente de donner quelques éléments de réponse à ce problème.

Un autre problème survient lors de la détection de périphériques identiques, c'est la distinction entre les deux. En effet, pour prendre un exemple que je côtoie, le robinet fourni par Oras, se nomme... Oras, et a une URL d'accès qui commence par ce même nom et est suivi des noms de capteurs qui sont naturellement identiques. Pour palier ce problème on peut notamment utiliser le concept ID@URI introduit par mon tuteur, Kary Främling qui consiste à combiner l'URI de l'objet à un identifiant unique.

4.4 Problème de localisation des acteurs

Une fois le périphérique détecté, si il est en mouvement, et que la disposition a une importance (par exemple si on cherche quelle personne est dans quelle pièce pour que la température lui soit idéale), on touche ici une nouvelle problématique de localisation. Comme on peut le lire dans [6], il existe plusieurs possibilités pour localiser un objet, on peut par exemple chercher à avoir une location relative grâce à un système de triangulation, ou une localisation absolue, c'est le cas des GPS ou des sols sensitifs. Quoi qu'il en soit, la localisation de personnes s'avère très compliqué, on peut par exemple imaginer deux personnes qui se croisent sur un sol sensitif, comment savoir que les deux personnes ont continué dans la même direction ou si elles ont fait un demi tour ? [12]

4.5 Action - Réaction

Une autre grande question se pose lorsqu'on décide de laisser la maison gérer elle même certains périphériques, c'est qu'elle ne réagirait peut-être pas comme

on le souhaiterait, ce qui entrainerait une sensation inverse que celle souhaitée : l'inconfort, ce qui serait un comble ! On pourrait penser que si l'application est bien configurée il pourrait ne pas avoir de problèmes, mais ce serait sans compter sur l'environnement changeant. Prenons un exemple pour illustrer le problème. On veut que la climatisation ne se déclenche que lorsque la température extérieur est supérieure à la température souhaitée, sinon il faut ouvrir les fenêtres. Mais que se passe-t-il si on ouvre grand les fenêtres et que notre voisin est en train de tondre la pelouse ? Ne pourrait-on pas être gêné par ce bruit alors que nous sommes en train d'écouter un merveilleux disque de Mozart ? Il en va de même, si nous invitons une personne allergique au pollen et que la fenêtre s'ouvre.

Les solutions sont pour le moment peu évidente, néanmoins quelques pistes semblent envisageables. Comme par exemple, considérer tous les objets comme des agents, ainsi lorsqu'un nouvel agent entre dans une zone, le système détecte ses croyances et peut ainsi décider d'agir différemment. (Il ne faut néanmoins pas que l'agent soit corrompu, en effet une compagnie qui vend les climatiseurs pourrait très bien se faire passer pour une personne allergique afin de faire laisser les fenêtres fermées alors que ce ne serait pas nécessaire). la programmation logique semble également une solution intéressante ; néanmoins, la solution la plus pratique reste de donner à l'utilisateur le choix final de la décision, de ne se servir de l'informatique que comme un outil d'aide à la décision. L'utilisateur peut ainsi voir sur un écran qu'il lui est conseillé d'ouvrir la fenêtre, mais c'est à lui de faire le choix. En fonction des choix de l'utilisateur, on pourrait même envisager un apprentissage des préférences afin de proposer les choix les plus judicieux.

[7]

Deuxième partie

Mon travail

Chapitre 5

Mise en place d'un serveur oBIX

5.1 Background

Le but de ce travail est relatif à l'automatisation dans la maison ainsi que la mise en réseau de différents ordinateurs.

L'automatisation dans les bâtiments Traditionnellement, on parle d'automatisation au sein d'un bâtiment pour les système de chauffage, ventilation et air conditionné (HVAC). La principale raison d'être de ce système est de rendre la consommation d'énergie plus efficace. Néanmoins, on peut également considérer un aspect de confort et de sécurité.

5.2 Qu'est ce que oBIX ?

Tout d'abord oBIX signifie "open Building Information eXchange"

oBIX est une initiative de grandes industries pour définir un langage XML standardisé, et une ligne de conduite pour web service afin de rendre possible la communication entre la partie mécanique, le système électrique et les applications d'un bâtiment. Le développement d'oBIX a débuté en avril 2003, et la version 1.0 a été révélé en 2006, avec en prime une boîte à outil permettant de manipuler les objets en JAVA.

De manière plus concrète oBIX est un langage XML permettant de classer l'ensemble des états d'un périphérique selon des types prédéfinies qui sont détaillés dans le schéma suivant :

Ainsi chaque données peut être illustrée d'une manière ou d'une autre par un document XML respectant la norme oBIX.

Côté réseau En plus de la représentation des données sous forme d'objets XML, oBIX définit un ensemble de règles à suivre pour communiquer via SOAP ou HTTP.

En ce qui me concerne je me suis focaliser sur le protocole HTTP, ainsi les quatre méthodes de requête avaient une action particulière :

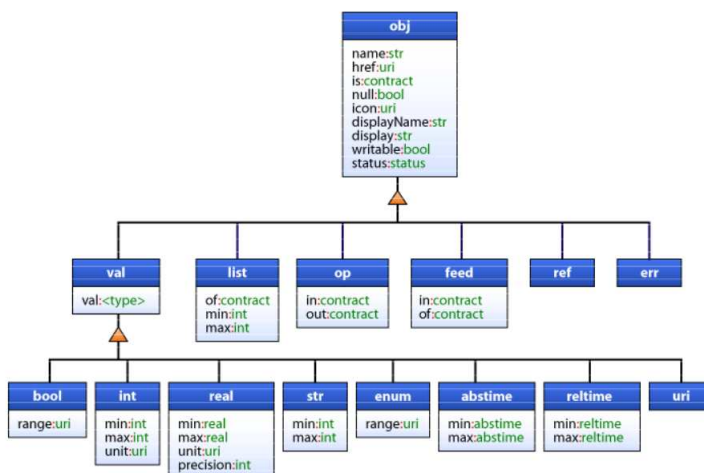
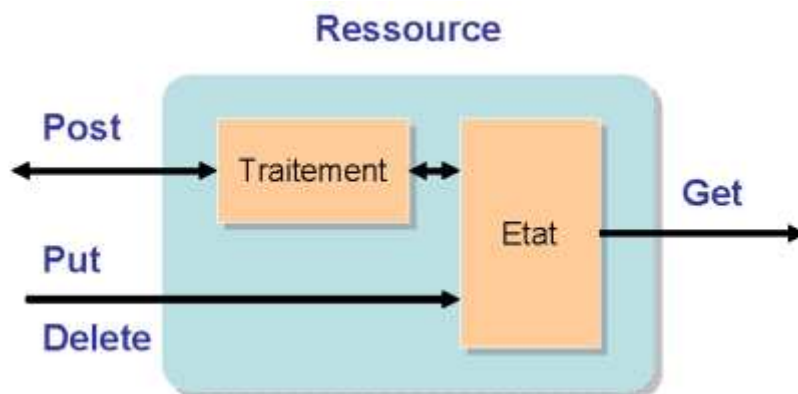


FIG. 5.1 – Représentation de la structure XML de oBIX

- GET permet de lire les informations se situant à l'URL indiquée
- POST permet d'invoquer des méthodes à distance
- PUT permet d'écrire
- DELETE qui supprime un élément.



5.3 Les concurrents de oBIX

Il existe plusieurs systèmes sur le marché. Les plus populaires utilisent les technologies X10, BACnet et LonWorks. X10 a été développé en 1975 et a été la première technologie à supporter les accès à distance. BACnet a vu son développement débuté en 1987 puis est devenu en 1995 ASHRAE/ANSI et devenait un standard ISO en 2003. Malheureusement, ce projet échoua car les différents fournisseurs ont créé des périphériques qui ne communiquaient pas ensemble. X10 et BACnet se concentraient principalement sur les systèmes HVAC, LonWorks prend également en considération la gestion de la lumière, la consommation d'énergie, la gestion de la sécurité. Malheureusement LonWorks était basé sur BACnet et les mêmes problèmes sont donc survenus.

On peut citer également en vrac Modbus,KNX-IP, Mbus, Z-Wave, OPC, SNMP.

5.4 Le serveur oBIX

Pour débiter mon travail j’ai commencé par mettre en place un serveur oBIX simple me permettant de me familiariser avec la norme et de comprendre comment agissaient les interactions entre les clients et le serveur. J’ai donc créé une servlet respectant les spécifications oBIX. Je me suis donc servi du document qu’on peut trouver en annexe afin de respecter scrupuleusement la marche à suivre. Tout d’abord la première chose à faire était de mettre en place des réponses spécifiques à certaines URL. Par exemple la page “About” est censé décrire la personne ayant développée le serveur pour quelle société et quelle est la version du serveur. La page d’accueil doit permettre d’accéder aux différentes méthodes comme le “WatchService” permettant d’accéder à la liste des changement d’états. Il fallait ensuite pour chaque requêtes GET faire une simple lecture des données, pour les requêtes POST, ce sont des invocations de méthodes et pour les requêtes de type PUT, ont pour but d’écrire les données.

Une fois ce mini serveur mis en place il était possible de lui envoyer des messages par PUT et d’activer ou désactiver des lampes virtuelles ainsi qu’au robinet Oras.

Ce serveur oBIX était le plus simple possible, c’est à dire qu’il ne sauvegardait pas réellement les valeurs envoyées par les différents périphériques, le but était ici de me familiariser avec la norme oBIX ainsi que ses méthodes de communication.

Plus concrètement, pour qu’un serveur oBIX fonctionne, il faut suivre un ensemble de règle définie dans les spécifications oBIX :

1. ReadLobby : Le périphérique récupère un ensemble d’adresses qui vont lui servir par la suite
2. SignUp : Le périphérique se signal au serveur en lui indiquant la structure des valeurs qui le compose
3. MakeWatch : Le périphérique récupère les adresses des méthodes à appeler pour recevoir les modifications et ajouter de nouvelles données
4. PollChange : Analyse la liste des modifications et met ses états à jour.

L’envoi d’informations, du périphérique au serveur, et inversement, s’effectuent à interval régulier via un thread.

Chapitre 6

Intégration de oBIX a DIALOG

6.1 Qu'est ce que DIALOG ?

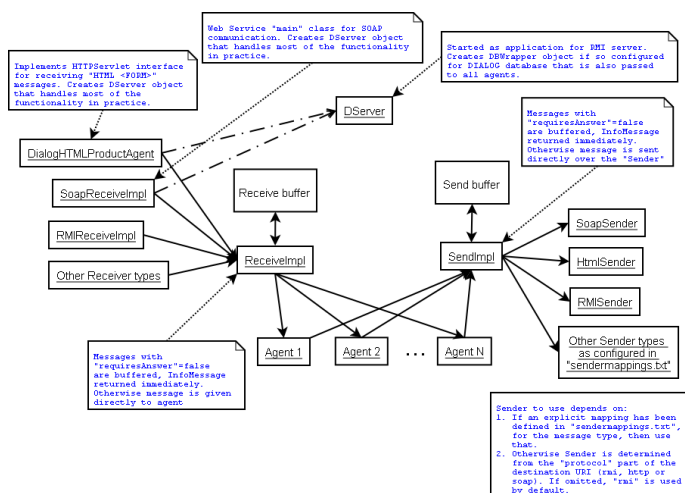
DIALOG signifie "Distributed Information Architectures for Collaborative LOGistics". C'est un système distribué développé en JAVA. DIALOG est compatible avec tous les systèmes d'exploitation et est déployé comme un WebService. Les principales applications industrielles de DIALOG sont :

- le suivi logistique
- gestion d'informations sur les produits
- coordination des informations entre les différents sites d'une entreprise

Du point de vue de la communication, DIALOG a essentiellement une architecture peer-to-peer (P2P). Comme dans tous les systèmes P2P, tous les noeuds DIALOG sont égaux, c'est à dire qu'ils peuvent tous envoyer et recevoir des messages. Par conséquent, tous les noeuds du réseau peuvent demander et envoyer des informations au sujet des parties dont il a des informations.

6.2 oBIX dans DIALOG

DIALOG a un mode de fonctionnement très simple qui est résumé par ce schéma :



Il m'a donc fallu créer un Receveur qui a comme intérêt d'écouter l'ensemble des émissions de messages vers le serveur. L'unique but de ce receveur était de récupérer les messages oBIX et de les encapsuler dans un DialogMessage. Une fois le DialogMessage créé le système DIALOG s'occupe d'acheminer le message vers le bon agent (en réalité, le système DIALOG envoie les messages à l'ensemble des agents en leur informant le type de message qui est envoyé et chaque agent a une liste de messages qu'il est capable d'interpréter, si ce message ne fait pas parti de cette liste il n'est simplement pas pris en compte par l'agent)

Par la suite il m'a donc fallu créer un Agent capable d'interpréter les messages oBIX, c'est donc à lui que revenait la tâche de respecter les spécifications oBIX. Par la suite, il m'a également fallu mettre en place la communication entre cet agent et la base de données. C'est en réalité le but principal du projet que j'avais à effectuer : en effet la structure générale de oBIX permet normalement de sauvegarder les données dans un fichier XML et lorsque de nouvelles données arrivent les anciennes sont détruites. Ici, j'ai donc repris la représentation du modèle oBIX pour créer mes tables de la base de données afin d'insérer chaque éléments dans la base avec toutes ses informations. Grâce à la sauvegarde de l'ensemble des données il est donc possible de dégager des tendances ou de faire des statistiques permettant d'améliorer les réactions des périphériques. Néanmoins, un problème pourrait se profiler en sauvegardant autant de données, surtout si une multitudes de périphériques se connectent, c'est le remplissage de la base de données, il faudrait donc mettre en place un système de nettoyage de cette base qui effacerait l'ensemble des vieilles valeurs.

6.3 La démonstration en place à Posintra

La force de Posintra, et plus particulièrement du département STOK est la présence d'une démonstration qui attire de nombreux visiteurs. Cette démonstration met en avant une communication entre différents périphériques permettant ainsi de rendre la maison intelligente. Tout d'abord un sol sensitif permet de détecter la présence de personnes dans la maison ainsi que leurs positions. Ainsi si le sol détecte une présence alors que personne n'est censé être là une alarme se déclenche et un texto est envoyé au propriétaire de la maison. Il existe une communication entre le radiateur et une machine de ventilation. Ainsi le chauffage ne fonctionnera pas lorsque la ventilation est en marche. De la même manière la ventilation tentera de se déclencher au maximum lors des absences des occupants de la maison. L'ensemble de ces paramètres est naturellement modifiable via Internet et un simple navigateur. La démonstration intègre également des lampes controlables via un simple téléphone, un média center, une prise détectrice de consommation anormale d'électricité. Cette démonstration est appelée à être installée chez mon tuteur dans le but de collecter de véritables données lors d'une utilisation réelle. Ces données permettront ainsi d'affiner les réglages et de montrer comment ce système peut améliorer la vie des occupants d'une maison tout en permettant une économie d'énergie non négligeable.

Conclusion et perspectives

Le problème principal de l'infomatique diffuse est la communication entre l'ensemble des systèmes. En effet, sans communication tous les autres problèmes ne seraient même pas envisageable. Certaines normes tentent de s'imposer comme oBIX dans le domaine de l'automatisation de bâtiments. C'est pourquoi il m'a été demandé de l'intégrer au cours de mon stage, dans le système DIALOG.

Cette intégration n'est malheureusement pas complètement finie, en effet si la partie communication gérée par fichiers est fonctionnelle, celle par base de données est encore en phase de développement.

Au-delà de la partie développement, ce stage m'a permis de me poser quelques questions relatives à ce domaine et qu'on pourrait envisager comme perspectives.

Par exemple, il serait intéressant de voir comment réagirait le système en cas de coupure du serveur (bug, panne d'électricité,...) En effet, dans mon cas, lors de mes tests, si un problème survenait alors que le robinet était en train de couler, le robinet restait alors allumer tant que je n'avais pas redémarré le serveur. Il serait peut-être intéressant de mettre en place un système permettant de basculer automatiquement en mode manuel lorsque qu'un crash survient.

Un autre point reste également en suspend, c'est la présence de deux périphériques du même type. Si dans ma démonstration, j'insère un deuxième robinet identique, le serveur ne saura pas comment les différencier. Le système ID@URI proposé par Kary Främling pourrait s'avérer être une solution, l'URI étant déjà mis en place, il ne faudrait que rajouter un identifiant unique à chacun des produits pour pouvoir les différencier.

J'ai également pu me poser la question des préférences utilisateurs, bien que ce ne soit pas vraiment un cas évident avec l'utilisation d'un robinet, néanmoins, considérons que ce robinet soit celui situé dans les toilettes, il pourrait être intéressant que la température de l'eau soit réglée en fonction de chacun des utilisateurs. Avec cette question des préférences on pointe également le problème de localisation des individus.

Enfin, un des problèmes les plus important à mon sens, est comment pourrait-on mettre en place un système permettant de détecter directement les périphériques se connectant au serveur. Dans le cas de mon robinet, il faut exécuter un adaptateur qui permet au serveur de connaître la présence de ce périphérique.

Malgré l'ensemble de ces questions sans réelles réponses, on peut tout de même imaginer une intégration intéressante de oBIX. D'un point de vu concret, il serait possible de coupler le sol sensitif au serveur et lorsque une personne se trouve dans une zone autoriser le déclenchement du robinet.

Annexe A

Présentation de l'entreprise

A.1 Posintra Oy

Posintra Oy est une compagnie de développement de business située à Porvoo dans la région de l'Est Nyland placée sous la direction de Fredrick von Schoultz. Sa principale raison d'être est de faciliter le développement et la structuration de petites et moyennes entreprises. Posintra vient juste de finaliser a large projet qui consuste à créer une base de données de toutes les compagnies de la ville de Porvoo (Cette base de donnée est consultable sur le site Internet de Posintra : <http://www.posintra.fi>) Cette base de donnée offre à l'ensemble des utilisateurs un accès simple aux informations sur les compagnies présnet dans la zone de Porvo, mais Posintra est aussi active sur de grand projets de développements qu'on peut classer dans trois catégories :

- Une activité de services
- Un service d'automatisation de bâtiments
- Un service permettant de promulguer certains évènements touristiques.

A.2 STOK

STOL est un département de Posintra, c'est un centre de développement qui a pour but de fournir des compétences dans les services de gestion de bâtiments. STOL fait parti du développement de Posintra dans la ville de Porvoo.

STOK joue le rôle de cataliseur et tente de regrouper l'industrie, les universités et la communauté de chercheurs dans des projets ayant pour sujet l'automatisation des bâtiments. Les activités de STOK peuvent se diviser selon quatres catégories :

- Technologie : STOK développe oSSG, un an open Source Service Gateway. oSSG est une plateforme open source implémentant l'interface oBIX développée par OASIS. Tous les documents et développements sont publics et disponibles gratuitement sous licence GPL.

Un laboratoire tentant d'intégrer le développement de oSSG est en cours de construction. Le laboratoire contient des équipement donnés par des fournisseurs soutenant les activités de STOK. Ce laboratoire est ouvert au public pour des visites.

- Innovation : STOK essaye également de regrouper les utilisateurs, les fournisseurs de technologie, les chercheurs et la communauté universitaire. Le but est de trouver de nouvelles solutions aux problèmes réels des utilisateurs. via un ensemble de projet. Ensemble de projets qui peuvent être commercialisés par l'industrie
- Economie d'énergie : Un des principaux projets de STOK, est la création de projet en relation avec les économies d'énergie. Outre la communication inter-périphériques présentée dans le laboratoire, la construction d'un lotissement de maisons pensées dans le but d'utiliser au maximum les énergies renouvelables et utilisant également une forte isolation est mise en place.
- Education : STOK soutient la communauté éducative en fournissant une plateforme d'intégration oSSG qui peut être utilisée à des buts éducatifs. Au cours du printemps 2007, STOK a fourni des cours relatifs aux équipements intégrés dans le laboratoire. STOK offre régulièrement la possibilité à des étudiants de rédiger leurs mémoires de master.

A.3 Le personnel

STOK regroupe huit personnes :

- Leena Tuomi, la directrice du projet
- Kari Pulkkinen, ingénieur dans les systèmes électriques
- Kary Främling, mon tuteur et responsable de projet à mi-temps (son autre mi-temps, il le passe à l'université de Helsinki)
- Åsa Nystedt, experte en création de bâtiments respectueux de la nature
- Jan Nyman, ingénieur en développement logiciel et responsable de projet
- Kariuki Mbotha, ingénieur en développement logiciel
- Petri Sallinen (qui ne travaille qu'à mi-temps à STOK)
- Jemina Tuominen, assistante

A.4 Les partenaires

A.4.1 Enervent

Tout d'abord, il est à noter que la société Enervent est connue sous le nom Venténer en France pour des raisons qu'on peut aisément comprendre. Depuis plus de 25 ans, Venténer fabrique et commercialise les systèmes de traitement d'air chaud ou froid. Leurs bureaux et leur usine sont situés dans les mêmes locaux que Posintra, c'est eux qui ont fourni au département STOK la ventilation présente dans la démonstration.

A.4.2 Nokia

La principale compagnie finlandaise, Nokia, un des principaux acteurs de téléphonie mobile s'est également lancée dans la création d'un centre de contrôle de maisons. Depuis peu, Nokia a créé une société nommée *There*. et dont l'ensemble des informations sont disponibles sur <http://www.therecorporation.com>. Un lien s'était créé entre Nokia et STOK, il perdure avec la création de cette

nouvelle société ainsi, STOK peut bénéficier en avant première du HCC de Nokia.

A.4.3 Ensto

Ensto, est également une société finlandaise créant des armoires électriques. De nombreuses armoires présentes et autres accessoires électriques sont mis à la disposition de STOK.

A.4.4 Oras

Oras développe, produit et vend des robinetteries, des valves industrielles et des modules. Cette compagnie a fourni à STOK un robinet remplie de capteurs ainsi qu'un adaptateur oBIX, c'est la raison pour laquelle nous devons intégrer la norme oBIX au serveur en place.

Annexe A

Extra

A.1 Le séminaire UBI Summit 2009

J'ai eu la chance de participer à un des grands événements de l'informatique diffuse en Finlande : l'UBI Summit. Lors de cette journée je n'ai malheureusement pu assister qu'à la moitié des séances car une partie de la conférence était en finnois. Néanmoins, assister à cette conférence a été une excellente expérience.

<http://www.oske.net/ubi/>

A.1.1 Première intervention : Hanna Marttinen-Deakins

Le but de cette première intervention était de faire un état des lieux de l'informatique diffuse en Asie et plus particulièrement au Japon, lieu où ce domaine est largement avancé.

Tout d'abord, Hanna Marttinen-Deakins nous a présenté un système appelée "Location experience", le but est d'obtenir un maximum d'informations sur un lieu grâce à son iPhone et de pouvoir rajouter ses propres informations. Ensuite elle nous a présenté un projet qui se déroule à Singapour, où on équipe des poissons de puces RFID afin de vérifier leurs déplacements et la vitesse d'extinction afin de gérer aux mieux les ressources de nourriture. Nous avons continué par une entreprise nommée Haier qui tente d'insérer l'informatique diffuse dans différents endroits via ses produits U-Home, U-Hotel, U-Office... un système de contact de réfrigérateurs via SMS est déjà mis en place par exemple (vous pouvez par exemple contacter votre réfrigérateur pour savoir s'il vous reste un produit lorsque vous êtes en train de faire vos courses.) Elle nous a également présenté un système de paiement par téléphone déjà en action depuis de nombreuses années au Japon qui marche grâce à de nombreux partenariats notamment avec McDonald. Enfin, la présentation d'un paquetage de sécurité pour enfant a ponctué la présentation, les parents sont automatiquement prévenu lorsque leur enfant quitte une zone définie.

A.1.2 Deuxième intervention : Nicholas R. S. Evans

Nicholas R. S. Evans est le vice-président de SRI Consulting Business Intelligence et nous a présenté les résultats de ses études sur les avancées des

différents projets relatifs à l’informatique diffuse et aux énergies renouvelables. Il en ressort six domaines en pleine activité :

- <http://www.sric-bi.com>
- Le charbon propre
- Les matériaux capables de stocker de l’énergie
- Les biocarburants
- Les moyens d’extensions de la vie
- Les robots personnels
- L’informatique des choses

A.1.3 Troisième intervention : Ryan Genz et Francesca Rosella

Enfin la dernière présentation était un peu plus exotique donc je ne m’étendrai pas sur le sujet, pour exemple, voici deux produits amusants voir intéressants :

- Le premier produit, n’est pas le plus intéressant, il s’appelle “Hug shirt”, il est couvert de capteur de chaleur et de systèmes permettant de faire des vibrations. Ainsi, il est possible de faire des câlins à distance.
- Le deuxième est appelé “Mobile phone dress”, c’est une robe qui joue le rôle de téléphone, ainsi on est sûr de ne manquer aucun appel et cela permet également d’améliorer le design des vêtements en ayant la possibilité de supprimer les poches tout en conservant son téléphone.

A.2 La journée de Posintra

Lors de mon stage a eu lieu la journée de Posintra. Cette journée avait pour but de présenter aux nouveaux venus les différentes activités de l’entreprise et de chercher à lier les différentes activités des différents départements. Cette journée se déroulant en finnois, mon tuteur a décidé de profiter de celle-ci pour me faire visiter l’université de technologie de Helsinki (TKK) ainsi que le laboratoire de puce RFID.

A.2.1 La visite de l’université de Helsinki

Je me suis donc rendu le jeudi 4 juin au sein de l’Université située à Espoo avec Kariuki M’Botha. Nous avons tout d’abord rejoint Andrey Litvinov qui nous a fait visiter les locaux puis nous a représenté brièvement son projet qui était très proche du mien : son but était d’intégrer la norme oBIX au serveur de Nokia présenté précédemment. Au cours de cette visite Andrey nous a également présenté un autre projet de l’université appelé XIDE qui a pour but de créer des interfaces graphiques automatiquement à partir de sources XML ce qui rend l’interface complètement indépendante du langage utilisé comme moteur d’application.

A.2.2 La “Design Factory”

Nous avons ensuite visité la “Design Factory”, ce lieu a été créé dans le but de regrouper les objets pensés dans l’université de technologie, tout en intégrant le modèles sociaux développés à l’université de sciences humaines et enfin en se

servant des capacités marketing de l'université de finances de Helsinki. Dans ce lieu on peut voir de nombreuses salles de réflexions mais également quelques débuts de projets. Les plus intéressants étaient :

- Une table permettant de recharger les appareils électroniques (notamment téléphones portables, assistants personnels,...) lorsqu'ils sont posés sur elle.
- L'élaboration d'un magasin intelligent grâce notamment à une intégration de puce RFID

A.2.3 Le RFID Labs

Pour finir cette journée de Posintra, Kary nous avait organisé une visite très intéressante du laboratoire RFID situé à Helsinki-Vantaa. La présentation nous a été faite par Sami Kalliokoski. Malheureusement ce laboratoire était en reconstruction, nous n'avons donc pas pu voir les puces RFID en action dans un milieu concret. Néanmoins, après une explication de ce que sont les puces RFID et une petite démonstration de petit produit en développement, notamment la création d'une puce RFID particulière permettant de tracer les batteries de téléphone. Ce projet a été demandé par Nokia dans le but d'identifier quelles sont leurs batteries et de pouvoir les récupérer dans un cadre d'un recyclage. Les puces RFID ayant du mal à fonctionner dans deux types de cas que sont la présence de liquide et la présence d'éléments métalliques, il a donc fallu adopter un moyen pour contrer ce problème. Ici a été mise en place une bague métallique autour de la puce RFID permettant de couper les interférences de la batterie. Puis, nous avons abordé un sujet plus proche de mes centres d'intérêts avec les futurs installations qui seront mise en place au sein du laboratoire. La première démonstration qui sera mise en place est celle qui m'intéresse le plus mais aussi, malheureusement, celle qui sera le moins développé. Il s'agit en effet d'une pièce de maison intelligente où chacun des objets sera marqué par des tag RFID ce qui permettrait par exemple de savoir où les objets sont situés dans la maison. Le deuxième projet est un magasin intelligent, le projet est plus avancé que le précédent, en effet, hormis la possibilité de situer les différents produits dans la boutique, ce qui facilitera grandement les inventaires, la mise en place d'un miroir affichant des informations relatives aux vêtements essayés par le client. Les exemples donnés par Sami étaient :

- Affichage des tailles restantes pour le produit
- Affichage d'articles pouvant être associé au produit essayé (par exemple une sélection de cravates d'un coloris adéquat à la chemise essayée)
- Affichage de produits similaires pour faciliter le choix à l'utilisateur

On pourrait même imaginer, que la personne puisse sélectionner sur le miroir tactile un des éléments indiqués ci-dessus et qu'ils soient directement apportés au client.

Les tags RFID pourraient également servir d'anti-vol et si quelqu'un cherche à dérober un produit on pourrait mettre en place un système de voix de synthèse annonçant quel produit est en train d'être dérobé.

Bibliographie

- [1] Keith W. Edwards and Rebecca E. Grinter. At home with ubiquitous computing : Seven challenges. In *UbiComp '01 : Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 256–272, 2001.
- [2] W. Keith Edwards. Discovery systems in ubiquitous computing. In *IEEE Pervasive Computing*, pages 70–77, 2006.
- [3] Jérôme Euzenat, Jérôme Pierson, and Fano Ramparani. Dynamic context management for pervasive applications. *Knowledge engineering review*, 23(1) :21–49, 2008.
- [4] Margaret Fleck, Marcos Frid, Tim Kindberg, Eamonn O'Brien-Strain, Rakhi Rajani, and Mirjana Spasojevic. From informing to remembering : Ubiquitous systems in interactive museums. In *IEEE Pervasive Computing*, volume 1, pages 13–21, 2002.
- [5] Kary Främling and Jan Nyman. Information architecture for intelligent products in the internet of things. In *Beyond Business Logistics. Proceedings of 20th NOFOMA Logistics Conference*, pages 221–232, 2008.
- [6] Jeffrey Hightower and Gaetano Borriella. Location systems for ubiquitous computing. In *IEEE Computer*, pages 57–66, 2001.
- [7] W. Jouve, N. Ibrahim, L. Réveillère, F. Le Mouël, and C. Consel. Building home monitoring applications : From design to implementation into the amigo middleware. In *Proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA'2007)*, 2007.
- [8] Michael Klein, Andreas Schmidt, and Rolf Lauer. Ontology-centred design of an ambient middleware for assisted living : The case of soprano. In Thomas Kirste, Birgitta König-Ries, and Ralf Salomon, editors, *Towards Ambient Intelligence : Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU)*, 30th Annual German Conference on Artificial Intelligence, 2007.
- [9] Carsten Magerkurth, Richard Etter, Maddy Janse, Juha Kela, Otilia Kocsis, and Fano Ramparany. F. : An intelligent user service architecture for networked home environments. In *Proc. of the 2nd Int. Conf. on Intelligent Environments*, pages 5–6.
- [10] Gerben G. Meyer, Kary Främling, and Jan Holmström. Intelligent products : A survey. volume 60, pages 137–148, 2009.

- [11] Joseph Newman, Martin Wagner, Martin Bauer, Asa Macwilliams, Thomas Pintaric, Dagmar Beyer, Daniel Pustka, Franz Strasser, Dieter Schmalstieg, Gudrun Klinker, and Technische Universität Wien. Ubiquitous tracking for augmented reality. In *In Proc. of International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 192–201, 2004.
- [12] Nirmalya Roy, Abhishek Roy, and Sajal K. Das. Context-aware resource management in multi-inhabitant smart homes : A nash h-learning based approach. In *Pervasive Computing and Communications, IEEE International Conference on*, pages 148–158, 2006.
- [13] Debashis Saha and Amitava Mukherjee. Pervasive computing : a paradigm for the 21st century. In *IEEE Computer*, pages 25–31, 2003.
- [14] M. Satyanarayanan. Pervasive computing : vision and challenges. In *Personal Communications, IEEE [see also IEEE Wireless Communications]*, pages 10–17, 2001.
- [15] Frank Stajano. Security for ubiquitous computing. 2002.
- [16] Mathieu Vallée, Fano Ramparany, and Laurent Vercoeur. Composition flexible de services d'objets communicants. In *UbiMob '05 : Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, pages 185–192, 2005.
- [17] Upkar Varshney. Network access and security issues in ubiquitous computing. 2001.
- [18] Mark Weiser. The computer for the 21st century. In *Scientific American*, volume 265, pages 66–75, 1991.
- [19] Mark Weiser. The future of ubiquitous computing on campus. In *Commun. ACM*, pages 41–42, 1998.