Software Engineering for Internet of Things

Résumé

C’est en 1968 au cours de la conférence NATO que Douglas McIlroy introduit pour la première fois la notion de composant logiciel. Ce paradigme de programmation promeut la réutilisation massive et la fabrication de logiciel par assemblage de composants. Mais aujourd’hui, le domaine des composants logiciel ne tient pas encore ses promesses en terme de réutilisabilité et de portabilité des composants. En effet, il est souvent nécessaire de créer un code de glue pour permettre l’assemblage de composants existants. Par conséquent, il est souvent moins coûteux de réimplanter un composant plutôt que de réutiliser des composants existants. En outre la construction de composants par compositions d’autres composants n’est pas de mise dans tous les modèles.

Avec la démocratisation des technologies dans l’habitat (domotique), de nouveaux besoins en terme de génie logiciel se font à la foi présents et pressants, pour faire face aux problématiques d’hétérogénéité des technologies utilisées, d’évolutivité du logiciel, d’adaptation à son environnement au cours de son exécution, et bien d’autres. Afin de répondre au plus juste aux besoin de chacun, et aux spécificités de chaque installation pour faire face à des problématiques allant du handicap aux économies d’énergies, les systèmes logiciels de gestion et de contrôle de ces installations doivent être fabriqués rapidement et se doivent d’être totalement fiables, et doivent être personnalisables par l’utilisateur lui-même.

La thèse défendue dans ce document prend le parie que l’utilisation conjointe d’un modèle de composant logiciel contraint, inspiré par l’électronique, et d’un environnement d’exécution ouvert et non contraint, permet d’adresser les problématiques d’interopérabilité, d’évolutivité, d’adaptation et de fiabilité, en ouvrant le champs des possibles dans le domaine des technologies de l’habitat et de l’utilisation de ces dernières pour l’assistance à domicile de personnes dépendantes.

EnTiMid, réalisation logicielle de cette thèse, s’appuie sur les paradigmes de programmation orientée services et à base de composants, enrichis par une approche utilisant les modèles à l’exécution, méthode de développement logiciel dirigé par les modèles. S’inspirant fortement de l’électronique, EnTiMid permet nativement l’interopérabilité, la réutilisabilité et offre une intégration aisée de produits domotiques et de services disponibles à travers Internet.

L’adéquation d’EnTiMid avec les problématiques de terrain a été validée par son utilisation dans le domaine de la domotique au service de la personne, et plus précisément, dans le contexte de l’assistance à domicile aux personnes âgées ou dépendantes au cours d’un projet de la métropole Rennaise.

Summary

Table des matères

I) Principles and Concepts 4

A) Introduction to MDE 4

B) Component models 4

C) Service Oriented Architectures 4

D) Internet Of Things 4

i) What is IoT? 4

ii) Where is it used? 5

iii) Why is it more complex than automation? 5

iv) Plethora of devices 5

v) Why SE is not ready to address this field? 5

vi) Standard vs. Private Protocols 5

E) Internet Of Services 5

F) The Cloud 5

II) Context of the work, identification of requirements 5

A) Home Automation 5

i) Pistes de développement dans le monde 5

ii) Pistes de développement en Europe 5

iii) Cas de la france 6

B) The AAL Context 6

C) Relevant problems 6

i) Interoperability 6

ii) Opening 6

iii) Adaptation 6

iv) Evolution 6

v) Variability Management 6

vi) Remote Control 6

vii) Distribution 7

viii) Safety & Security 7

ix) User profiling 7

x) Acceptability 7

D) Existing Approaches 7

i) Agent Based 7

ii) Service Based 8

iii) Component-Based 8

E) Generalization and synthesis 8

i) Problems addressed in this thesis 8

III) Thesis and realization 8

A) Bases 8

B) Naïve 8

C) Using M@RT 8

D) EnTiMid component model 8

i) Details of the model 9

ii) How it addresses requirements 9

iii) 9

IV) Architecture Synthesis 9

A) Static: FD, CT, CI 9

B) Dynamic; MAD Editor, BEPL 9

C) Synthétiseur, Compilateur 9

V) Validation 9

A) A travers différents points de vue 9

B) IDA 9

C) ITI Project 9

D) Industrial contacts 9

E) Public Demonstrations 9

i) FDLS 9

ii) Inauguration ESIR 9

iii) AERES 9

F) Spin Off 9

VI) Conclusion 9

A) Synthesis board 10

B) Pros&Cons 10

C) IoT-IoS 10

VII) Perspectives 10

A) Open Control/Command operating system ^^ 10

B) EUP 10

C) FuzyLogic 10

VIII) Bibliography 10

IX) Glossary 10

X) Definitions 11

# Principles and Concepts

Le domaine de l’informatique, et plus particulièrement du genie logiciel, est un monde dans lequelle les choses évoluent vite, s’empilent et s’entrecroisent, essayant de rendre jour après jour les logiciels plus fiables et rapides à concevoir. Dans cette section sont rapportés les principaux concepts necessaries à la compréhension de la problématique levée et du travail realisé présentés dans les sections suivantes.

## Introduction to MDE

La thèse s’est déroulée au sein de l’Equipe TRISKELL, équipe dont le cœur de métier est l’ingénierie des modèles. De fait, l’ensemble de cette thèse repose sur des fondements issus de l’ingénierie des modèles. Cette section fait le point sur ces fondations.

## Component models

Au dessus de ces fondations, la notion de composant logiciel est apparue. Il est essentiel d’en saisir les concepts, car la thèse défendue dans ce document apporte une évolution dans ce domaine.

## Service Oriented Architectures

Le domaine du génie logiciel est un domaine actif. Faisant suite aux modèles de composants, le paradigme d’architectures orientées services apparaît comme un axe intéressant pour le développement d’applications autour de l’Internet en général.

## Internet Of Things

L’Internet des objets désigne une approche dans laquelle les objets de tous les jours atteignent un niveau de maturité suffisant pour leur permettre d’interagir les uns avec les autres et rendre des services à valeur ajoutée. Etant encore dans ses prémices, cette approche se cherche encore des outils logiciels de développement pour décrire les interactions, et les services rendus. Il est fort probable que la solution réside en un amalgame subtil entre les architectures à composants et celles à services.

### What is IoT?

\cite{ITU-IOT-Ex.Sum-2005}

The Internet Of Things originates from the Auto-ID Labs[[1]](#footnote--1), successor of the MIT Auto-ID Center. This lab is an independent network of seven academic research centers that “research and develop new technologies for revolutionizing global commerce and providing previously un-realizable consumer benefits”. They associated to architect the Internet Of Things together with EPCglobal.

EPCGlobal[[2]](#footnote-0) is a suite of standards (such as Electronic Product Code – EPC) and services from GS1 designed to support the development and use of the RFID technology in today ICT.

A couple of definitions can be found on Internet for this term. The first, found on Wikipedia says: “

### Where is it used?

\cite{Gautier:2010}

### Why is it more complex than automation?

\cite{Gautier:2010}

### Plethora of devices

Santé, Confort éclairage, Confort Chauffage, Ouvrants, Téléphone Mobile,

### Why SE is not ready to address this field?

### Standard vs. Private Protocols

Even if some consortiums of academics and industrials bith in America or Europe try to collaborate to design a standart communication way, there will probably never exist a unified standard.

## Internet Of Services

A l’image de son compère, l’Internet des Services promeut l’interaction de services disponibles à travers Internet. L’orchestration d’un ensemble de services unitaires, rend possible la création de services plus complexes, à grande valeur ajoutée, et ce à moindre coût.

## The Cloud

Le concept de cloud détache les services et les objets d’une localisation precise. Autrement dit, les services et objets actifs sont diffus dans l’environnement d’exécution, et utilisables sans avoir à connaitre exactement leur localisation.

# Context of the work, identification of requirements

Les travaux présentés dans ce document s’efforcent d’utiliser les technologies et techniques de développement de pointe pour répondre à une problématique sociétale ; celle de l’assistance et du maintien à domicile (MAD) de personnes dépendantes. Afin d’en comprendre les enjeux, cette section brosse un panorama des préoccupations dans ce domaine, aboutissant à une spécifications des exigences en terme de génie logiciel.

Face à ces exigences, un sondage des solutions apportées par les technologies existantes nous donnera un ensemble de bonnes propriétés et de lacunes à combler pour répondre à l’ensemble des préoccupations du MAD.

## Home Automation

### Pistes de développement dans le monde

Corée => Jeux videos

Japon => Salle de bain (faire couler un bain depuis son telephone

Etats-unis => Sécurité des biens

### Pistes de développement en Europe

Italie => Design intérieur

Allemagne

Royaume Uni

### Cas de la france

Economies d’énergies

AAL

## The AAL Context

The aim of this section is to present the AAL context of this thesis. The description of this domain helps in understanding the requirements and challenges identified in the next section.

## Relevant problems

This section identifies the relevant problems/requirements software engineering should address to be able to face the next generation of software for internet of things, internet of services or ubiquitous systems.

### Interoperability

L’interconnexion des réseaux est la première exigence qui ressort d’une analyse des préoccupations des acteurs du domaine. De plus en plus, la nécessité de faire communiquer entre eux des appareils non compatibles (caméras, interrupteurs, capteurs de présence, etc.) en provenance de multiples constructeurs est au coeur des préoccupations. En effet, un service local à un habitat peut exiger le pilotage d’équipements d’une grande variété, aussi bien sur le plan fonctionnel que technologique. Cet impératif se retrouve également pour certains services globaux ayant pour objet la gestion au niveau d’une agglomération, d’un parc préexistant d’équipements potentiellement très hétérogènes. En conséquence, la conception de services locaux ou globaux repose sur l’orchestration de dispositifs en provenance de constructeurs différents. Or, à ce jour, les constructeurs tendent plutôt à vouloir capter leurs clients au travers de protocoles propriétaires incompatibles.

### Opening

L’ouverture à des protocoles de haut niveau semble être la deuxième exigence. Le modèle économique pressenti autour de la domotique se compare souvent à celui d’un Amazon ou d’un Google, il est difﬁcile de prévoir à l’avance tous les services qui pourront être construits par dessus ce type de système. Celui-ci doit donc permettre de développer et déployer facilement des services de haut niveau construits comme une agrégation de services de plus bas niveaux accessibles sur le web ou dans une maison.

### Adaptation

### Evolution

### Variability Management

### Remote Control

Favoriser les accès nomades représente la troisième exigence. L’accès aux services de la domotique ne doit plus se faire uniquement via une télécommande universelle localisée dans l’habitat. Les équipements doivent pouvoir être sollicités et administrés depuis plusieurs lieux (la personne depuis son habitat, l’inﬁrmière depuis sa voiture, le médecin depuis son cabinet, le technicien depuis son bureau, etc.), au travers de différents matériels (une station de travail, un assistant personnel, un téléphone intelligent, une télévision, etc.), selon des interfaces adaptées à chaque contexte et type d’intervenant (depuis un navigateur web, une application dédiée, un client de messagerie instantanée, etc.). Cela implique, en particulier, le support de protocoles de communication de haut niveau facilitant l’écriture de services nomades.

### Distribution

Des solutions dynamiques de déploiement favorisant la prise en charge des des nouveaux dispositifs et technologies à venir est bel et bien la quatrième exigence. Les trois exigences pré- cédentes font l’hypothèse d’un univers ﬁgé en termes de services, de matériels et de protocoles, mais il est clair que de nouveaux protocoles vont être proposés ou standardisés, de nouveaux services plus riches et plus complexes vont être publiés. Il est donc vital qu’une telle plate-forme de domotique permette des évolutions sans perturber les composants logiciels qu’elle héberge.

### Safety & Security

La sûreté et la sécurité forment enﬁn la cinquième et dernière exigence. Les systèmes domotiques déployés devront garantir des niveaux de ﬁabilité, de disponibilité, de sûreté (sécurité des biens et des personnes) et de sécurité (contrôle des accès aux services selon le type des intervenants) élevés. En particulier, un fonctionnement en mode dégradé doit permettre de maintenir un niveau de service adéquat, malgré la présence de pannes de dispositifs matériels, de composants logiciels, ou la perte d’une connexion réseau.

### User profiling

#### EUP

### Acceptability

Un système acceptable est un système dont on comprend les résactions.

Les gens doivent s’approprier la technologie

La technologie doit être capable de s’adapter aux utilisateurs

## Existing Approaches

This section evaluates different technical approaches to solve the problems, through a survey on pros and cons of each approach with relation to the requirements listed in previous section.

### Agent Based

#### UBIWARE

\cite{Terziyan:2009}

### Service Based

#### WS

#### REST

#### OSGi

### Component-Based

#### SCA

#### Fractal

#### Connector generation

## Generalization and synthesis

### Problems addressed in this thesis

Interoperability

Opening

Adaptation

Variability Management

Evolutivity

Usability / understandability for everyone

# Thesis and realization

Les bases conceptuelles posées, les exigences remontées par le terrain et l’étude crituque des réponses apportées par les approches existantes permettent d’expliciter la these défendue dans ce document.

Après une presentation conceptuelle de l’approche, cette section décrit et detaille les options d’implementation envisages, puis rapporte la realisation finale.

## Bases

Se rapprocher du modèle électronique

## Naïve

Faire des objets conformes, avec API included

## Using M@RT

Faire des objets conformes avec API included et les gérer avec un modèle

## EnTiMid component model

Faire des objets conformes API excluded, managed by model containing APIs

### Details of the model

### How it addresses requirements

### 

# Architecture Synthesis

## Static: FD, CT, CI

## Dynamic; MAD Editor, BEPL

BPEL distribué. Pas encore adressé mais résolu par nous….

## Synthétiseur, Compilateur

# Validation

Tout au long de la realisation de ces travaux, des validations tant au niveau technique qu’humain ont été réalisées. Il est essentiel, dans le contexte du MAD, que les solutions avancées et proposes par cette these répondent aux besoins du domaine, et soient acceptées par les multiples utilisateurs autour de ce système. Cette section fait echo des divers retours d’expériences et validations menées sur la solution proposée.

## A travers différents points de vue

* Dévelopeur
* Technicien/Installateur
* Utilisateur final

## IDA

## ITI Project

## Industrial contacts

## Public Demonstrations

### FDLS

### Inauguration ESIR

### AERES

## Spin Off

# Conclusion

A partir d’une etude précise des besoins dans le contexte societal fort du maintien à domicile de personne dependant, et d’un socle de connaissances en domotique et IDM, la these présentée dans ce document offre une solution tirant partie des bonnes pratiques d’approches existantes, et combles les manques, afin de répondre au plus juste aux problématiques du domaine.

## Synthesis board

Comparison between listed existing approaches and mine

## Pros&Cons

## IoT-IoS

Seamless integration of IoT and IoS

# Perspectives

Au dela du travail realise, il reste aujourd’hui de nombreauses pistes à explorer autour de ces travaux. Cette section donne un apercu des possibles suites à donner dans ce domaine.

## Open Control/Command operating system ^^

## EUP

Make it possible for everybody to understand and modify software behaviour

## FuzyLogic

Base behaviour reflexions on non-fixed value such as “high” or “low”. This could incite people to use the system and appropriate the configuration.

# Bibliography

* + Using MDE to build a schisophrenic middleware

# Glossary

DPWS : Device Profile for Web Services

EPC : Electronic Product Code™

EUP : End User Programming

ICT : Information and Communication Technology

IOS : Internet Of Services

IOT : Internet Of Things

MDE : Model Driven Engineering

RFID : Radio Frequency IDentification

UPnP : Universal Plug & Play

MAD

HAD

# Definitions

Plateforme : Environnement logiciel et materiel pleinement spécifié dans lequel s’exécute une application.

Plateforme d’exécution : Evironnement materiel pure prévu pour accueillir des applications.

Environnement d’exécution : Support d’exécution logiciel pour une apllication finale.

1. http://www.autoidlabs.org/ [↑](#footnote-ref--1)
2. http://www.gs1.org/epcglobal/ [↑](#footnote-ref-0)