# RESTFUL FRAMEWORK FOR COLLABORATIVE INTERNET OF THINGS BASED ON IEC 61850

JORGE PARRA MOLINA

eman ta zabal zazu

Universidad del País Vasco — Euskal Herriko Unibertsitatea

Supervisor: Eduardo Jacob Taquet
Department of Communications Engineering
Faculty of Engineering of Bilbao
University of the Basque Country (UPV/EHU)

February 2016

Jorge Parra Molina: *RESTful Framework for Collaborative Internet of Things based on IEC 61850,* Industrial Engineer,



SUPERVISORS:
Eduardo Jacob Taquet

LOCATION:
Bilbao

TIME FRAME:
February 2016

# ABSTRACT

Smart environments where envisioned by Mark Weiser in 1991 when he formulated the Ubiquitous Computing paradigm, defining them as physical locations populated by devices, highly integrated in the environment, with identification, sensing and actuation capabilities. Ambient Intelligence (AmI) was the first evolution of this paradigm, with a user-centric view of the ubiquitous services. Nowadays, Internet of Things (IoT) is the next evolution of these two concepts, expanding the scope of the individual devices, represented as *things*, from a local environment to the *Internet* as global network.

For implementing these scenarios, one of the biggest challenges in this research area is to establish automatic mechanisms to compose, in a dynamic way, services on demand. The objective of the composition is to satisfy global needs combining the existing services. This thesis proposes a collaborative architecture among the *things* in the IoT domain.

SOAP/XML based Web Services are suitable for IoT scenarios where these services can be published, discovered and invoked dynamically. Alternatively, REpresentational State Transfer (REST) has gained much attention from the community and is considered as a lighter and cleaner technology for communication in the *Internet*. Despite such benefits of REST, the dynamic discovery and eventing of RESTful services are yet considered a major hurdle to get the full potential of REST based approaches. This thesis addresses this issue, by providing a RESTful discovery and eventing specification.

Another aspect addressed by this thesis is the application level organization of the distributed *things*. Electrical power systems and substations are one of the most representative domains of networked individual *things*. The most prominent and adopted solution for standardizing the information models in these domain is the International Electrotechnical Commission

(IEC) 61850 standard. However, the communication protocols defined by the standard are not completely adequate for IoT. Thus, this thesis elaborates on the suitability of IEC 61850 information models for IoT applications proposing a RESTful mapping of the abstract services.

Finally, dependability related issues are addressed when considering the application of IoT architectures in critical domains such as health care or safety critical systems. This thesis analyzes the communication aspects from various message transfer perspectives. The cornerstone of the proposed approach is the consideration of a dependable RESTful communication throughout the message trail in the IoT environment.

# RESUMEN

En 1991, Mark Weiser formuló el paradigma de Computación Ubicua definiendo el concepto de Entorno Inteligente como un espacio físico repleto de dispositivos, muy integrados en el entorno, y con capacidades de identificación, sensorización y actuación. La Inteligencia Ambiental (InAm) fue la primera evolución de este paradigma, centrando completamente los servicios ubicuos en los usuarios. Internet de las Cosas (IoT) expande el ámbito de localización de estos dispositivos y servicios ubicuos, representados como *cosas*, de un entorno local a *Internet* como red global.

Para la implementación de estos escenarios de aplicación, la colaboración entre las *cosas* es uno de los principales retos de investigación. El objetivo de esta colaboración es ser capaces de satisfacer necesidades globales mediante la combinación de servicios individuales. Esta Tesis propone una arquitectura colaborativa entre las *cosas* desplegadas en *Internet*.

Las tecnologías alrededor de los Servicios Web SOAP/XML, adecuadas para IoT, soportan aspectos claves para un sistema colaborativo como la publicación, descubrimiento, control y gestión de eventos de los dispositivos. Como alternativa, REST ha ganado terreno en este ámbito por ser considerada una opción más ligera, sencilla y natural para la comunicación en *Internet*. Es un paradigma de arquitectura que representa los servicios como *recursos* publicados y consumidos mediante los protocolos en los que se basa la Web. Sin embargo, no existen protocolos para descubrimiento y gestión de eventos para recursos REST. Esta Tesis aborda dicha carencia proponiendo una especificación de estos protocolos para arquitecturas REST.

Otro aspecto importante es la representación, a nivel de aplicación, de las *cosas* distribuidas. Las redes de subestaciones eléctricas son un dominio de aplicación muy representativo de sistemas de *cosas* interconectadas. Entre

las propuestas para la estandarización de los modelos de información y comunicación en este dominio que podrían aplicarse, de manera similar, a IoT, destaca el estándar IEC 61850. Sin embargo, los protocolos de comunicación definidos por el estándar no son adecuados para IoT. Por lo tanto, esta Tesis analiza la idoneidad del IEC 61850 para escenarios IoT y propone un protocolo de comunicación REST para sus servicios.

Por último, se trata la problemática asociada a la confiabilidad que debe proporcionar una arquitectura *IoT* para dominios de aplicación relacionados con la salud o sistemas de seguridad funcional (Safety). Esta Tesis analiza los aspectos de comunicación desde diversas perspectivas de transferencia de mensajes y propone mecanismos confiables para la comunicación REST en el entorno IoT.

# LABURPENA

Mark Weiserrek *Nonahiko Konputazio*aren paradigma definitu zuen 1991n. Ikuspegi honen barruan *Ingurumen adimentsua* deritzon kontzeptua aurreikusi zuen. Kontzeptu honek inguruarekin guztiz integratuta eta bertan identifikatu, sentitu eta eragiteko gai diren gailuez beteriko espazio fisikoak definitzen ditu. Ingurumen Adimena (InA) izan zen paradigma honen lehen eboluzioa, nonahiko zerbitzuak erabiltzaileen beharretara bideratuz. Gauzen Interneta (GI)k zerbitzu eta gailuen esparrua zabaldu egiten du gertuko ingurune batetik sare globalago batera, *Internet*era, zerbitzu eta gailuak *gauza* bezala irudikatuaz bertan.

Eskenatoki teknologiko hauek aurrera eramateko ikerketa munduan dagoen erronkarik handienetariko bat zerbitzuak eskatu ahala automatikoki eta modu dinamikoan konposatzeko mekanismoak ezartzea da. Konposaketa honen helburua balio erantsi handiagoko beharrak asetzea da, banakako zerbitzuen konbinaketaren bidez. Tesi honek *Internet*en hedaturiko *gauz*en elkarlanerako arkitektura bat proposatzen du.

SOAP/XML Web Zerbitzuetan oinarritutako teknologiak GI inguruak garatzeko egokiak dira, zerbitzuen argitalpen, jakinarazpen, bilaketa eta kontrola ahalbidetzen dituztelako. Halaber, alternatiba gisa, REST teknologia gero eta garrantzi gehiago hartzen ari da, arinagoa, errazagoa eta *Internet*eko komunikazioetarako egokiagoa delako. Hala ere, azken honek ez du bilaketa eta jakinarazpen dinamikorako protokolorik eskaintzen. Tesi honek gabezia hauek jorratzen ditu eta REST arkitekturarako protokolo hauen espezifikazio bat proposatzen du.

Landu beharreko beste alderdi garrantzitsu bat bananduriko *gauz*en aplikazio mailako antolakuntza da. Azpiestazio elektrikoak elkar konektaturiko sistemen aplikazio domeinu oso adierazgarritzat kontsidera daitezke. Domeinu mota hauetarako eredu eta komunikazioa estandarizatzeko proposamenen

artean IEC 61850 nabarmentzen da, nahiz eta estandar honek definituriko komunikazio protokoloak ez diren GI-erako egokiak. Hori dela eta, Tesi honek IEC 61850aren egokitasuna aztertzen du GI-erako eta REST proposatzen du komunikazio protokolo gisa.

Azkenik, osasun eta segurtasun funtzionaleko esparruetarako garatutako GI arkitektura batek eskaini behar duen fidagarritasunaren arazoa jorratu da. Tesi honek mezuen transferentzia ikuspuntutik komunikazioetako hainbat alderdi aztertzen ditu eta GI inguruneetako REST bidezko komunikazioetarako mekanismo fidagarriak proposatzen ditu.

# RESUMEN EJECUTIVO

En 1991, Mark Weiser formuló el paradigma de Computación Ubicua definiendo el concepto de *Entorno Inteligente* como un espacio físico repleto de dispositivos, muy integrados en el entorno, y con capacidades de identificación, sensorización y actuación. La InAm fue la primera evolución de este paradigma, centrando completamente los servicios ubicuos en los usuarios. Internet de las Cosas (IoT) expande el ámbito de localización de estos dispositivos y servicios ubicuos, representados como *cosas*, de un entorno local a *Internet* como red global.

Para la realización de la visión de IoT se deben tener en cuenta una serie de aspectos que diferencian estos sistemas de las soluciones que se han aplicado tradicionalmente a *Entornos Inteligentes*. Una arquitectura de aplicación para un sistema IoT debe tener en cuenta diferentes aspectos:

- Mecanismos de interacción entre las *cosas* que permitan construir aplicaciones mediante la combinación de las funcionalidades que ofrecen cada una de ellas por separado.
- Modelos estandarizados de representación de las *cosas*: para uniformizar la organización y estructuración de la información que proporciona cada una de las cosas, con el fin de facilitar la interoperabilidad a la hora de construir aplicaciones.
- Protocolos de comunicación adecuados a *Internet* como red de *cosas*: la World Wide Web (WWW) es una arquitectura que sigue los principios establecidos por REST. Por lo tanto, para tener un encaje adecuado en esta red, tanto los mecanismos de interacción, como los modelos de representación deben estar orientados a una arquitectura de este tipo.

Para la implementación de estos escenarios de aplicación, la colaboración entre las *cosas* es uno de los principales retos de investigación. El objetivo de esta colaboración es ser capaces de satisfacer necesidades globales mediante la combinación de servicios que proporciona cada *cosa* de forma individual.

Frente al modelo centralizado en el que existe un coordinador único que orqueste la ejecución de los servicios independientes, se propone un modelo de composición descentralizado, en el que cada una de las *cosas* es autónoma y sabe cómo debe actuar por si misma ante los cambios en el contexto del sistema. Por lo tanto, las reglas de comportamiento de cada una de las *cosas* están distribuidas entre todos los dispositivos, siendo cada uno de ellos el responsable de determinar su comportamiento. Esta solución favorece las escalabilidad del sistema, puesto que permite la incorporación de nuevas funcionalidades sin afectar al resto. Además, con la ausencia de un controlador central, se elimina la problemática derivada del punto de fallo único (Single Point of Failure), aumentando las disponibilidad del sistema.

Para ello, además de que cada uno de los elementos de la red sea capaz de conocer su comportamiento, cada una de las *cosas* en la red debe ser consciente de la existencia de otras *cosas*, y ser capaz de recibir información de ellas. Puesto que hablamos de entornos móviles dinámicos en los que los dispositivos y servicios pueden aparecer o desaparecer de la red, es necesario definir:

- Un protocolo de descubrimiento que permita que las *cosas* se anuncien en la red describiendo sus funcionalidades, y que puedan buscar otras *cosas* en las que estén interesadas.

- Un protocolo para la gestión de notificaciones que permita que las *cosas* se suscriban a eventos producidos por otras, de forma que reciban la información de su interés.

Esta Tesis propone, por una parte, una arquitectura colaborativa entre las *cosas* desplegadas en *Internet* describiendo un modelo de lógica distribuida. Por otra parte, define protocolos de descubrimiento y gestión de eventos para habilitar esta colaboración entre *cosas* independientes.

## MODELO DE INFORMACIÓN UNIFORME

Otro aspecto importante es la representación, a nivel de aplicación, de las *cosas* distribuidas. Para favorecer la interoperabilidad de *cosas* a la hora de construir aplicaciones, es necesario uniformizar y estandarizar la descripción estructural de la información que proporciona cada una de las cosas.

Las redes de subestaciones eléctricas son un dominio de aplicación muy representativo de sistemas de *cosas* interconectadas. Entre las propuestas para la estandarización de los modelos de información y comunicación en este dominio que podrían aplicarse, de manera similar, a IoT, destaca el estándar IEC 61850. Esta especificación define un modelo de referencia genérico para representar cualquier tipo de elemento (*cosa*) que puede existir en una subestación, de forma que facilita la interoperabilidad entre dispositivos heterogéneos. Además, el estándar IEC 61850 define un conjunto de servicios concretos para interactuar con el modelo de información genérico para la descripción, monitorización y control (como servicios principales) de los dispositivos en las subestaciones.

Es un modelo sencillo, escalable y genérico que puede aplicarse a cualquier otro dominio de aplicación y que, a grandes rasgos, estructura la información de la siguiente forma:

- Logical Device: representa e identifica a un dispositivo que está formado por un conjunto de nodos lógicos (Logical Nodes). Como ejemplo se puede considerar un transformador de una subestación eléctrica.

- Logical Node: representa e identifica una funcionalidad concreta de cada dispositivo que proporciona y agrupa un conjunto de datos (Data) que permiten caracterizar esa funcionalidad. Por ejemplo, el sistema de refrigeración del transformador.

- Data: representa cada uno de los datos concretos en forma de un conjunto de atributos (DataAttributes). Por ejemplo, la temperatura del sistema de refrigeración.
- DataAttribute: representan el valor de cada uno de atributos que describen un dato concreto. Por ejemplo, el valor instantáneo de la temperatura, sus límites máximo y mínimo admisibles, la propia calidad del valor instantáneo, etc.

Se podría adoptar esta misma idea para representar de forma uniforme las *cosas* en *Internet*. Aplicando este mismo modelo de referencia para describir de forma estándar las *cosas* se podrían utilizar los mismos servicios para interactuar con ellas. El IEC 61850 define, además, mapeos de esos servicios a protocolos de comunicación concretos. Sin embargo, los protocolos de comunicación definidos por el estándar no son adecuados para IoT.

Por lo tanto, esta Tesis analiza y verifica la idoneidad del IEC 61850 para escenarios IoT.

## PROTOCOLOS DE COMUNICACIÓN

Tanto para los protocolos de descubrimiento y gestión de eventos, como para la representación del modelo de información y los servicios para la interacción con este modelo, se debe tener en cuenta que el ámbito de las aplicaciones IoT es *Internet*. Es por ello que hay que considerar que las soluciones a aplicar sean adecuadas para ello.

Las tecnologías alrededor de los Servicios Web SOAP/XML, adecuadas para IoT, soportan aspectos claves para un sistema colaborativo como la publicación, descubrimiento, control y gestión de eventos de los dispositivos. Como alternativa, REST ha ganado terreno en este ámbito por ser considerada una opción más ligera, sencilla y natural para la comunicación en *Internet*. Es un paradigma de arquitectura que representa los servicios como *recursos* publicados y consumidos mediante los protocolos en los que se basa la Web. Sin embargo, no existen protocolos para descubrimiento y gestión de eventos para recursos REST. Esta Tesis aborda dicha carencia proponiendo una especificación de estos protocolos para arquitecturas REST.

Respecto al tema de representación de los dispositivos, la estructuración de la información en base a *recursos* propuesto por REST frente a *servicios* encaja perfectamente con el modelo de referencia de información propuesto por la norma IEC 61850. La composición de Uniform Resource Identifier (URI)s que identifican los recursos en la Web como concatenación de nombres (por ejemplo, para identificar el límite máximo de temperatura para el sistema de refrigeración de un autotransformador se podría considerar la URI `/Autotransformador/Refrigeracion/Temperatura/LimiteSuperior`), que se corresponde de forma directa con la identificación de los elementos en el modelo de referencia IEC 61850 (`LD.Ln/Data/DataAttribute`). Del mismo mode se podrían adoptar los mecanismos de interacción con los recursos en la Web (principalmente mediante el protocolo Hypertext Transfer Protocol (HTTP)) para el mapeo de los servicios de la norma IEC 61850.

Esta Tesis propone aplicar los principios de REST, aplicables en *Internet* por ser una arquitectura REST, para los protocolos de descubrimiento y gestión de eventos, así como para el modelo de referencia y los servicios de la norma IEC 61850.

Para ello se propone una estructuración de los correspondientes modelos de información en *recursos* REST, y se especifica el mapeo de los servicios para la interacción con el modelo de información mediante el protocolo HTTP aplicándolo a la manipulación de la representación de los recursos mediante los métodos defininidos en el protocolo HTTP:

- GET: para obtener la representación de un recurso concreto.

- PUT: para crear o modificar la representación de un recurso concreto.
- POST: para enviar, a un recurso concreto, información a no relacionada con su propia representación.
- DELETE: para eliminar un recurso concreto.

### CONFIABILIDAD

Por último, se trata la problemática asociada a la confiabilidad que debe proporcionar una arquitectura IoT para dominios de aplicación relacionados con la salud o sistemas de seguridad funcional (Safety). La característica principal de este tipo de sistemas es que se debe detectar cualquier falta o error que tenga como consecuencia una situación no segura. De esta forma, el sistema podrá actuar en consecuencia.

En un entorno altamente distribuido como éste, en el que fluye información desde los dispositivos sensores hasta los actuadores pasando por posibles elementos analizadores y procesadores de esa información, existe un gran número de mensajes que se intercambian entre las distintas *cosas*. Para conseguir construir aplicaciones confiables es necesario asegurar la transmisión de esta información, aplicando técnicas y medidas para detectar errores y faltas que puedan aparecer en el envío y recepción de estos mensajes. Para ello se identifican las posibles causas que puedan dar lugar a errores debidos a la transimisión de mensajes y se proponen mecanismos para detectar estos fallos. Puesto que se ha adoptado REST como arquitectura general, estos mecanismos se implementan siguiendo sus principios para la transmisión de mensajes.

Esta Tesis analiza los aspectos de comunicación desde diversas perspectivas de transferencia de mensajes y propone mecanismos confiables para la comunicación REST en el entorno IoT.

Research is to see what everybody else has seen, and to think what nobody else has thought.

— *Albert Szent-Györgyi*

An expert is one who knows more and more about less and less until he knows absolutely everything about nothing.

— *Nicholas Murray Butler*

Highly organized research is guaranteed to produce nothing new.

— *Frank Herbert*

# ACKNOWLEDGMENTS


Firstly, I would like to express my sincere gratitude to my advisor Prof. Eduardo Jacob for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

My sincere gratitude also goes to Prof. Abdulmotaleb El Saddik and Dr. Valérie Issarny, who provided me an opportunity to join their teams at University of Ottawa and INRIA Paris-Rocquencourt Research Centre, respectively, and who gave access to the laboratory and research facilities. Without they precious support it would not be possible to conduct this research.

I thank my colleagues in IK4-Ikerlan for the stimulating discussions, for the brilliant ideas, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last years.

It is with immense gratitude that I acknowledge the support and help of my friend Dr. M. Anwar Hossain for pushing me during all these years researching together.

Last but not the least, I would like to thank my family for supporting me throughout writing this thesis and my life in general.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LISTINGS

# ACRONYMS

**AAL** Ambient Assisted Living

**ACSI** Abstract Common Service Interface

**ADL** Activities of daily living

**AI** Artificial Intelligence

**AMI** Ambient Intelligence

**AMP** Ambient Media Preference

**API** Application Programming Interface

**BPEL4WS** Business Process Execution Language for Web Services

**BRCB** Buffered Report Control Block

**DAML** DARPA Agent Markup Language

**DER** Distributed Energy Resources

**DPWS** Device Profile for Web Services

**DVB** Digital Video Broadcasting

**ECA** Event, Condition, Action

**ECF** Executable Choreography Framework

**ECL** Executable Choreography Language

**FAME** Framework for Ambient Media

**FAN** Field Area Network

**FSM** Finite State Machines

**GI** Gauzen Interneta

**GENA** General Event Notification Architecture

**GOCB** GOOSE Control Block

**GOOSE** Generic Object Oriented Substation Event

**GPRS** General Packet Radio Service

**GSCB** GSSE Control Block

**GSM** Global System for Mobile communications

**GSSE** Generic Substation State Event

**GUI** Graphical User Interface

**HTML** HyperText Markup Language

**HTN** Hierarchical Task Network

**HTTP** Hypertext Transfer Protocol

**HVAC** Heating, ventilating, and air conditioning

**INA** Ingurumen Adimena

**IEC** International Electrotechnical Commission

**IED** Intelligent Electronic Device

**IMDB** Internet Movie Database

**INAM** Inteligencia Ambiental

**IOT** Internet of Things

**ISO** International Organization for Standardization

**IT** Information Technologies

**JSON** JavaScript Object Notation

**LAN** Local Area Network

**LCB** Log Control Block

**LCW** Local Closed World

**LD** Logical Device

**LN** Logical Node

**MD5** Message-Digest Algorithm 5

**MIME** Multipurpose Internet Mail Extensions

**MMS** Manufacturing Message Specification

**MSVCB** Multicast Sample Value Control Block

**OSGI** Open Services Gateway initiative

**OWL** Ontology Web Language

**PDU** Protocol Data Unit

**P2P** Peer to Peer

**RA** Research Activities

**RCB** Report Control Block

**REST** REpresentational State Transfer

**ROA** Resource Oriented Architecture

**RFID** Radio Frequency IDentification

**RG** Research Goals

**RTP** Real-time Transport Protocol

**RTSP** Real Time Streaming Protocol

**SCSM** Specific Communication Service Mappings

**SDK** Software Development Kit

**SG** Setting Group

**SHOP** Simple Hierarchical Ordered Planner

**SLP** Service Location Protocol

**SGCB** Setting Group Control Block

**SOA** Service Oriented Architecture

**SOAP** Simple Object Access Protocol

**SSDP** Simple Service Discovery Protocol

**SAWSDL** Semantic Annotations for WSDL

**SWSF** Semantic Web Services Framework

**SWSL** Semantic Web Services Language

**UDP** User Datagram Protocol

**UMTS** Universal Mobile Telecommunications System

**UPNP** Universal Plug & Play

**URCB** Unbuffered Report Control Block

**URI** Uniform Resource Identifier

**USVCB** Unicast Sample Value Control Block

**UWB** Ultra Wide Band

**V2G** Vehicle-to-Grid

**WAN** Wide Area Network

**WS** Web Service

**WSDL** Web Service Description Language

**WSMO** Web Service Modeling Ontology

**XML** eXtensible Markup Language

# 1 INTRODUCTION

This chapter presents and introduces the context of this thesis. In section 1.1, the scope and the different aspects that motivate this work are outlined: building smart and dependable environments with the integration of distributed devices that collaborate in the *Internet*.

The research objectives targeted by this work are described in Section 1.2:

- Studying the requirements of the communications for these environments to extend the scope of the devices from local spaces to the *Internet*.
- Analyzing the suitability of the IEC 61850 standard for this domain.
- Studying the requirements for dependable communication.

In order to reach these research goals, the set of research activities to be carried out are detailed in Section 1.3, which come up with the specification of a RESTful framework for collaborative systems that considers the discovery, eventing and dependability related aspects, adapted to the IEC 61850 standard.

Section 1.4 summarizes the thesis statement and lists the contributions resulted from this research work.

Finally, section 1.5 describes the structure of the whole document.

## 1.1 MOTIVATION

This research work is motivated by some open issues and challenges in the realization of smart environments according to the IoT paradigm, as described in this section.

### 1.1.1 Building smart environments

*'Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence'*. This is the scenario envisioned by Mark Weiser in 1991, creating the Pervasive or Ubiquitous Computing [105] paradigm. A smart environment, thus, can be defined as a physical location populated by devices with identification, sensing and actuation capabilities.

This first vision was extended by the concepts, ideas and scenarios applied to AmI [1] where the surrounding devices and services had a user-centric target, oriented to aid and support the users in the environment. Smart environments, according to AmI concepts, are electronically augmented surroundings with identification, sensor and actuator devices, and are aware of the presence of people to provide them timely and relevant services for better health and wellness support [83].

IoT is a new evolution of both concepts, focusing on the *spatial distribution* of the devices, as a set of interconnected *things* forming pervasive computing environments [61] and therefore, requiring enhanced communication protocols and distributed intelligence for the smart objects.

As also stated in [5, 61, 33], the IoT concept is the result of the intersection of three complementary visions:

- Internet-oriented vision: focused on the distribution and networking aspects, basically centered on middleware aspects. Its result should be a global network interconnecting smart objects by means of extended Internet technologies.
- Thing-oriented vision: pushing towards the integration of heterogeneous sensors and actuators. It considers

the set of supporting technologies necessary to realize such a vision such as Radio Frequency IDentification (RFID)s, sensor and actuators, machine-to-machine communication devices, and so on.

- Semantic-oriented vision: targeting the *semantic* representation of the devices and their capabilities, oriented to build applications and services, focusing on the knowledge derived from the information obtained from the system, leveraging such technologies to open new business and market opportunities.

A *thing* in the IoT vision could be defined as a physical entity with the following common features:

- It is uniquely identifiable and addressable in the whole system.
- It has computing capabilities.
- It has communication capabilities with other *things*.
- It has sensing or actuation capabilities which behave as interface with the real world.

Considering the spatial distribution aspects, the following challenging issues must be taken into account for the realization of the IoT idea[5]:

- **C1**. Full interoperability of interconnected devices: the inherent heterogeneity of technologies and capabilities of the devices and services to be integrated in such a globalized system requires both architectural (middleware based) and communication based solutions.
- **C2**. Higher degree of smartness: in a highly distributed environment, aiming at the collaboration among the devices, the centralized vision of a single control entity that orchestrates the whole system seems not to be very appropriate. Instead, moving some of the *intelligence* to the participant devices should help to achieve higher levels of availability, avoiding the single point of failure

of the centralized control entity. This objective can be taken into account, as more and more devices are equipped with increased networking and computation capabilities.

- **C3**. Scalability: considering the large scale of IoT environment and the dynamism of the *things* in the interconnected network.
- **C4**. Low computation and energy resources: both computing and energy consumption constraints become a relevant aspect in a massive deployment of spatially distributed devices.

The aim of this thesis is to study and explore these key aspects for building smart environments according to the IoT vision.

### 1.1.2 Enable dynamic integration of distributed things

There is a growing demand in designing a flexible architecture that aims to inter-operate among heterogeneous sensors, actuators and other services. The objective of developing such architecture is to support the users and applications in the smart environment such that they can access any offered functionality in a seamless fashion. A flexible architecture should seamlessly incorporate newly evolved devices and/or enable modifications to existing devices with minimal effort, despite the fact that the new or existing services may be tied to a different vendor with a vendor-specific interface[70].

Integration of heterogeneous devices for IoT requires a layered middleware [5] in order to minimize the development complexity. This is achieved abstracting the devices' functionalities and communication capabilities, providing a common set of objects with a well-defined interface. This common interface enables the aggregation and composition of the functionalities individually offered by each object, as well.

Several approaches and technologies have been proposed in order to address the above challenges, such as Open Services Gateway initiative (OSGi) [69], Device Profile for Web Services

(DPWS) [68, 55], Universal Plug & Play (UPnP) [87] and others. Among these, the Web Service (WS) family of protocols has gained much popularity for developing interoperable systems and applications, specially considering the benefits of supporting Service Oriented Architecture (SOA)s. However, it is considered heavier than what is required in the smart environment context and hence suffers from performance degradation.

As an alternative, REST has gained much attention from the community, which is also considered as a lighter and cleaner technology compared to the Simple Object Access Protocol (SOAP)-based web services [35].

REST is an architectural paradigm for distributed systems based on the *resource* concept, used for naming and identifying any kind of available information in the system. This kind of architecture must follow the next principles[28]:

- Unique identification of resources.
- Resources are described by representations.
- Uniform interface to interact with the resources.
- Stateless communication for applications.

Besides, it is simple to publish and use a RESTful web service [101]. As a result, more and more service providers are moving toward REST-based solution, which promotes a resource-centric model compared to a service-centric model.

Again, as for Intelligent Electronic Device (IED)s in section 1.1.4, the *resource* concept is not far at all from the *thing* concept in IoT. Hence, using RESTful principles for building IoT architectures seems to be coherent.

However, an adequate RESTful dynamic discovery for distributed *resources* or *things* is not resolved [9]. For smart environments, it is important to work on this discovery-related question because the addition or removal of a new *thing* needs to be propagated dynamically to all the other *thing* so that they can adjust their behavior. Another issue comes along with this discovery mechanism, which is the eventing (event management) because it is also an essential feature in a highly dynamic environment. Event management enables engagement and notifications among existing *things* for supporting functional collaboration and composition.

So, the following challenges can be identified in order to enable the RESTful integration of distributed *things*:

- **C5**. RESTful dynamic discovery mechanism for distributed *things*
- **C6**. RESTful event management for distributed *things*

This work is motivated by the need of specifying a lightweight middleware approach following the REST principles for enabling dynamic discovery and event management of networked resources.

### 1.1.3 Enable collaborative environments

Besides interoperability, *things* in smart environments should collaborate in order to augment, extend and improve the overall with respect the individual capabilities. The main limitation that exists to make real the smart environments is the little collaboration capability among the devices. If these devices are not capable of offering the expected or desired result, it might be possible that this result might be obtained by the collaboration among different devices [56, 85].

In order to achieve more flexibility and scalability, this collaboration among devices and services should follow a decentralized pattern where the peers could act autonomously on their behalf [107, 3].

This is also justified by the fact that more and more devices are equipped with increased networking and computation power, thereby providing the options of adopting a Peer to Peer (P2P)-like infrastructure. However, how to develop a P2P-like infrastructure composed of different *things* is still an issue.

The possibility of having P2P like infrastructure for distributed systems opens the door to investigate the distribution of application workflow logic on different peers. It will not only remove the burden of a centralized entity to maintain the complex application workflow to realize different scenarios, but also simplify the application workflow at a peer level, as each peer will be responsible to perform its own objective task depending on the environment notifications.

The issue of distributing the application workflow (effective decentralized service composition) logic to individual peers and show the flexibility it provides, is another motivation for this work: distribution of application workflow logic into the distributed peers to achieve simplicity and avoid single point of failure of a traditional centralized control entity.

For example, a lamp service peer may only be responsible to switch-on or switch-off the physical lamp device given the application logic it has. Moreover, having a true P2P-like architecture will enable the addition or removal of a service without affecting the overall operations. However, existing solutions are mostly geared towards a centralized control mechanism where the central entity is solely responsible to command how the individual device and service will react in response to various notifications from different devices and services.

Regarding the collaboration among *things*, the following challenges can be pointed out:

- **C7**. Functional composition of *things*
- **C8**. Decentralization of the application workflow among the distributed *things*

Therefore, the issue of distributing the application workflow (effective decentralized service composition) logic to individual peers and show the flexibility it provides, is another motivation for this work.

### 1.1.4 Applying IEC 61850 to smart environments

The idea of abstracting networked devices as individual *things* in order to build composed applications is, however, not novel in the industrial domain. IEC 61850 [41] is a part of the IEC's architecture for electrical power systems. Initially focused on providing an standardized reference for modeling, controlling and monitoring electrical substations, it's openness has extended its application range to new power system domains such as Wind Power Plants [42], Hydroelectric Power Plants [43] and Distributed Energy Resources (DER)s [46].

It considers the IED term referring to any device in a substation with processing capabilities that can share information and commands with other devices. Clearly, this concept shows great similarities with the *thing* idea in the IoT environment, and thus it could also be extended to the *Internet* domain.

IEC 61850 was designed and developed taking into account the following requirements [57]:

- High-speed IED to IED communication.
- Networkable.
- High-availability.
- Guaranteed delivery times.
- Standards based.
- Multi-vendor interoperability.
- Support for heterogeneous samples data.
- Support for File Transfer.
- Auto-configurable / configuration support.
- Support for security.

It must be emphasized that all of these requirements perfectly match and fit with other environments under the IoT definition out of electrical power systems and, thus, could be directly applied to them as well.

As IEC 61850 standardizes the principles for modeling and communication for electrical substation domain, aiming at the interoperability among the devices in the substation, the same requirements are adequate to propose IoT architectures. It proposes three well-defined aspects for modeling and interacting with the real world *things*.

- A simple information model for representing the diverse *things*.
- Abstract services to interact with the information model.
- Specific mappings of the abstract services to different communication protocols.

Both the information model and the abstract services can be totally applied to the IoT domain, while the current communication protocols proposed by the IEC 61850 are not completely adequate for IoT[49, 55]. The IEC defined a communication mapping based on SOAP-eXtensible Markup Language (XML) web services, when proposed the extension of the IEC 61850 scope from electrical substations to Wind Power Plants [45]. Therefore, this thesis elaborates on the suitability of IEC 61850 information models for IoT applications, extending it with a RESTful mapping of the abstract services.

Regarding this topic, the following challenges can be identified:

- **C9**. Adopt the IEC 61850 information model for IoT.
- **C10**. Define a communication mapping suitable for the *Internet*.

This work is motivated by the possibility of extending the range of application of IEC 61850 principles to the IoT domain.

### 1.1.5 Enable dependable environments

Among the several application domains for the IoT paradigm [5, 61], Ambient Assisted Living (AAL), due to its inherent relationship with healthcare, introduces a new aspect to be taken into account compared to other domains: dependability requirements.

AAL refers to technologically augmented environments that are sensitive and responsive to the presence of people and provide them support to maintain an independent and caring lifestyle [94, 83]. Sensing and alerting in response to an emergency situation is the key to such environments. Hence, an

AAL system receives various data from heterogeneous sensors and process it to understand the nature of a situation and take actions accordingly.

These AAL environments are oriented to offer caregiver's assistance in actions of Activities of daily living (ADL) and health monitoring. Nowadays, it is common in AAL facilities that there are several health care devices and sensors with varying level of computational and networking capabilities. As a result, it is possible to interconnect these devices in order to realize some distributed and collaborative applications for better healthcare support in AAL environments. However, service provisioning in such environments faces distinct challenges due to the heterogeneity of network, access technology, and sensing/actuation devices.

Situation analysis in AAL environment requires a lot of messages, which embeds sensors' data and processing outcomes, to be gradually transferred from the sensor module all the way to alert generator and response handler. It's a long pathway of communicating messages that change in content, format, type, and size when intended for different stakeholders. For the well-being of elderly people, it is critical to detect emergency situation and send alert messages in real-time [81]. Hence, AAL systems are considered as safety-critical systems that should provide safe, reliable and trusted messaging services when transferring sensor data and/or decisions from one unit to another.

Different alternatives exist when referring to sending messages in AAL environments, such as, SOAP-based Web Service [32] and REST [80] being the dominant technologies. While web service technology has become pervasive, REST proved to be an alternative lightweight technology with significant potential for service provisioning [9] and may be suitable for AAL communication. However, due to the safety-critical nature of AAL systems, the issues remain on how to transfer messages such that various message communication errors including data corruption, incorrect sequence, loss etc. can be minimized, while providing a lightweight solution.

The main challenges identified in this area are:

- **C11**. Define a dependable protocol to be applied in IoT.
- **C12**. Incorporate these dependability-related features to a RESTful architecture.

So, this work also addresses the variability of messaging requirements in an AAL environment and is motivated by the need of studying the dependability needs in terms of messaging for realizing AAL scenarios, and defining a proposal for that purpose.

## 1.2 RESEARCH GOALS

Taking into consideration the motivations and challenges described in previous section, this thesis has the following Research Goals (RG).

- **RG1**. Study the requirements of IoT paradigm related to building smart environments, considering a decentralized pattern for the resulting applications.
- **RG2**. Study the suitability of networking protocols applied to model and communicate with distributed devices to meet the IoT expectations for supporting dynamic environments of devices according to discovery and addressing needs.
- **RG3**. Study the suitability of applying the IEC 61850 modeling and interaction principles for representating the *things* in the IoT paradigm.
- **RG4**. Study the adequacy of the communication protocols defined for the IEC 61850 to the IoT networking and communication challenges.
- **RG5**. Study the requirements related to building dependable applications based on IoT.
- **RG6**. Define a framework for realizing and building IoT applications, extending and adapting the IEC 61850 definitions.

| Challenge | RG1 | RG2 | RG3 | RG4 | RG5 | RG6 |
|---|---|---|---|---|---|---|
| C1 | | X | | X | | X |
| C2 | X | | | | | X |
| C3 | X | X | | | | X |
| C4 | | | | X | | |
| C5 | | X | | | | X |
| C6 | | X | | | | X |
| C7 | X | | | | | X |
| C8 | X | | | | | |
| C9 | | | X | X | | X |
| C10 | | | X | X | | X |
| C11 | | | | | X | |
| C12 | | | | | X | X |

**Table 1.1:** Traceability of Challenges to Research Goals

Table 1.1 relates these research goals with the challenges identified in section 1.1.

## 1.3 PROPOSED APPROACH

In order to respond to the research goals described in section 1.2 the following Research Activities (RA) shall be conducted.

- **RA1**: Define a distributed framework for collaborative systems integrating surrounding devices.

  The proposal shall consider how the different *things* must be represented and act as individual peers in order to have a decentralized system, which is scalable by nature and avoids the single point-of-failure usually attributed to a centralized control entity. Further, it shall distribute the application workflow logic among the peers to develop a flexible architecture with autonomous behavior of the peers.

- **RA2**: Define a lightweight discovery and eventing protocol to enable decentralized communication.

  The proposed approach shall focus on specifying a lightweight middleware approach based on REST for enabling dynamic discovery and event management of networked resources. It shall define a RESTful mechanism so that the distributed *things* can discover one each other, and exchange notifications among them.

- **RA3**: Define a RESTful mapping for IEC 61850 model.

  This task shall define a RESTful communication mapping for IEC 61850 services, so that the information model and abstract services proposed by the IEC 61850 can be applied in the *Internet*.

- **RA4**: Define a safe communication pattern among distributed devices.

  This task shall analyze the messaging requirements for building dependable IoT scenarios, and propose a dependability-enabled RESTful communication pathway for alert-response management.

Table 1.2 relates these research activities with the research goals identified in section 1.2.

In order to test and validate the applicability of the proposed approaches in the IoT environment, different application domains shall be analyzed as case studies:

- Smart home scenario: to validate the suitability of a decentralized application logic among peers, as proposed in RA1.

- Ambient media selection scenario: to validate the discovery and eventing proposal defined in RA2.

| Research Goal | RA1 | RA2 | RA3 | RA4 |
|---|---|---|---|---|
| RG1 | X | | | |
| RG2 | | X | | |
| RG3 | | | X | |
| RG4 | | | X | |
| RG5 | | | | X |
| RG6 | X | X | X | X |

**Table 1.2:** Traceability of Research Goals to Research Activities

- Health care: to validate the dependability related information in the RESTful communication pattern described in RA4.

Further, analytical comparison of the proposed approaches with existing solutions shall be conducted, showing quantitative results to show the feasibility of the contributions:

- Comparison in terms of message size and processing time of the RESTful proposal for Discovery with UPnP and DPWS to validate the results of RA2.
- Comparison in terms of message size and processing time of the RESTful proposal for Eventing with UPnP and DPWS to validate the results of RA2.
- Comparison in terms of message size and processing time of the proposed RESTful mapping for the IEC 61850 with the the SOAP-XML Web Service mapping defined by the IEC 61400 to validate the results of RA3.
- Overhead of dependability related information in the communication protocol to validate the results of RA4.

## 1.4 THESIS STATEMENT AND CONTRIBUTIONS

The central idea of this Thesis is:

> RESTful mapping of IEC 61850, extended with discovery and eventing features, is a suitable approach to facilitate the realization of dependable IoT scenarios with a decentralized application control logic.
> A set of autonomous *things* located on the *Internet*, described following the principles stated by the IEC 61850, will be able to collaborate and perform composed actions, if they are augmented with dynamic discovery and eventing mechanisms.
> The RESTful protocol must be enhanced to support dependable communication, in order to realize safety critical IoT applications.

To this end, the Thesis has made the following contributions:

- Reference architecture for decentralized networks of devices based on a P2P approach.
- RESTful middleware for lightweight discovery, eventing and communication.
- RESTful mapping for IEC 61850.
- Analysis of dependability needs and impact in the IoT communication pattern, applied to RESTful communication.

## 1.5 THESIS ORGANIZATION

The remainder of this thesis is organized as follows: initially, a critical analysis of existing approaches regarding the thesis research goals is presented in section 2. Later, the resulting research contributions are detailed individually in section 3.

Section 4 lists the validation activities conducted to assess the proposed contributions. Finally, section 5 concludes the thesis summarizing the obtained results, showing the dissemination activity of the contributions of this thesis, and pointing out the future work in this area.

# 4 VALIDATION

In this chapter, we validate the contributions of this research work, considering two different perspectives. First, in section 4.1 we analyze and compare the proposed approach for RESTful discovery and eventing protocols with the discovery and eventing protocols defined for UPnP and DPWS. As the three of them are HTTP based protocols, the analysis and comparison is done based on the size of the messages that each protocol defines for:

- Advertisement of new services or devices in the network.
- Search for services or devices in the network.

We also analyze the overhead that is added when incorporating dependability-related information into the communication protocol.

Then, in section 4.2, a functional validation of the approaches is carried out. This validation is conducted checking the suitability and adequacy of the proposed solutions to different application domains. Some prototype implementations of the thesis contributions have been developed in order to check that they can definitively be applied to realize the thesis goal: building collaborative applications for IoT domain. The section describes the developed prototypes and the results of the implemented solutions.

## 4.1 ANALYTICAL VALIDATION

This section shows the results of the analysis and comparison of the approaches proposed in this thesis with existing solutions. The following aspects are considered in this analysis.

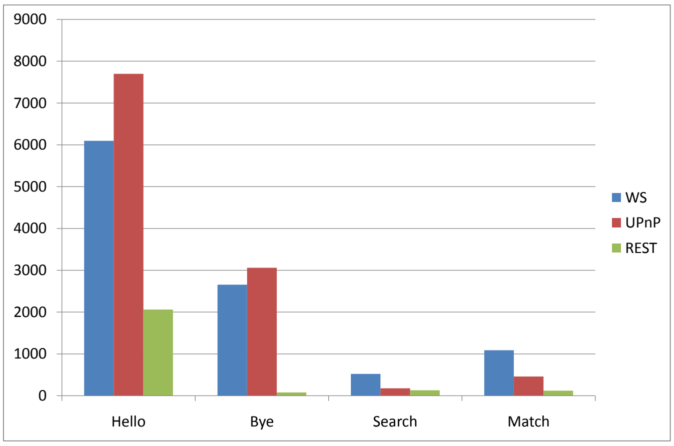


**Figure 4.34:** Advertising message size comparison

- RESTful discovery protocol is compared with SSDP and WS-Discovery.
- RESTful eventing protocol is compared with GENA and WS-Eventing.
- Overhead of including safety related information in the communication protocol.

### 4.1.1 RESTful discovery vs SSDP and WS-Discovery

This section analyzes and compares the size of the different messages that are involved in the discovery process for three different discovery protocols.

- SSDP
- WS-Discovery
- REST

Figure 4.34 shows the size of the different set of messages that are interchanged in the discovery use cases. There are four kind of messages:

- Hello: message sent by the resource, UPnP Device or Web Service when advertises itself in the network.
- Bye: message sent by the resource, UPnP Device or Web Service when is no longer available in the network. An example of the three different messages that illustrates the difference in size is provided in Listings 8, 9 and 10.
- Search: message sent by the resource client, UPnP Control Point or Web Service client when they want to search for a networked .resource, UPnP Device or Web Service.
- Match: message sent by the resource, UPnP Device or Web Service as response to the search message.

**Listing 8:** Bye message for WS

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws
    /2004/08/addressing" xmlns:d="http://schemas.xmlsoap
    .org/ws/2005/04/discovery" xmlns:s="http://www.w3.
    org/2003/05/soap-envelope">
        <s:Header>
                <wsa:To>urn:schemas-xmlsoap-
                    org:ws:2005:04:discovery</wsa:To>
                <wsa:Action>http://schemas.xmlsoap.org/
                    ws/2005/04/discovery/Bye</wsa:Action
                    >
                <wsa:MessageID>uuid:45da081b-5539-410d
                    -90a5-5d7d55d8c80b</wsa:MessageID>
                <d:AppSequence InstanceId="3289995071"
                    MessageNumber="2" />
        </s:Header>
        <s:Body>
                <d:Bye>
                        <d:EndpointReference>
                                <wsa:Address>
                                    uuid:ddb2e2dc-d02b-4
                                    a11-b977-55
                                    e1fdd567b8</
                                    wsa:Address>
                        </d:EndpointReference>
```

```
                </d:Bye>
        </s:Body>
</s:Envelope>
```

Clearly, the overhead introduced by using XML for encoding the message information in Listing 8 makes it significantly bigger and harder to parse than the UPnP and REST messages.

**Listing 9:** ByeBye message of UPnP

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age=1800
CONTENT-LENGTH:0
LOCATION: http://172.16.6.3:8080/siemensTestDevice/
    description.xml
NT: urn:schemas-upnp-org:service:SwitchPower:1
NTS: ssdp:byebye
USN: uuid:siemensTestDevice::urn:schemas-upnp-
    org:service:SwitchPower:1
```

Both UPnP and REST messages are follow the HTTP request structure while WS discovery messages are pure XML documents.

**Listing 10:** RESTful Bye message

```
DELETE /Resources/REST/lamps/lamp1 HTTP/1.1
Host: 224.0.0.1:8888
UUID: uuidlamp1
```

### 4.1.2 RESTful eventing vs GENA(UPnP) and WS-Eventing (DPWS)

This section analyzes and compares the size of the different messages that are involved in the eventing use cases for three different discovery protocols.

- GENA
- WS-Eventing

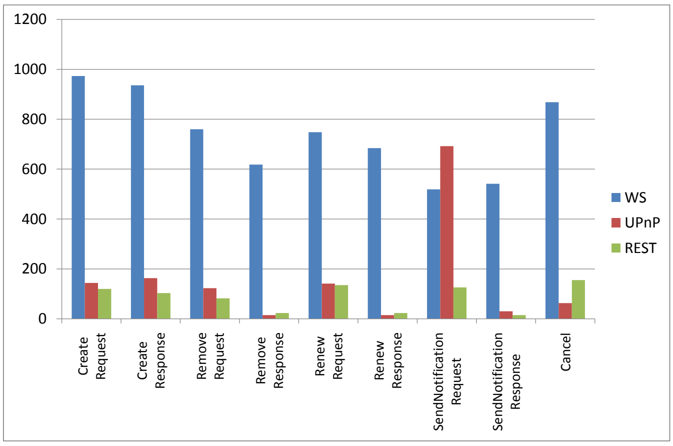


**Figure 4.35:** Eventing message size comparison

- REST

Figure 4.35 shows the size of the different set of messages that are interchanged in the eventing use cases. There are five kind of messages:

- Create subscription request and response: message sent by the resource client, UPnP Control Point or Web Service client to ask for subscribing to events produced by the resource, UPnP Device or Web Service, which responds to the request.
- Remove subscription request and response: message sent by the resource client, UPnP Control Point or Web Service client to stop the reception of events produced by the resource, UPnP Device or Web Service, which responds to the request.
- Renew subscription request and response: message sent by the resource client, UPnP Control Point or Web Service client to renew the event subscription.

- Send notification request and response: message sent by the resource, UPnP Control Device or Web Service to the subscribers containing the event data.
- Cancel request: message sent by the resource , UPnP Control Device or Web Service to the subscribers to inform that the subscriptions are no longer available.

Clearly, WS is the heaviest protocol for alamos all kind of eventing messages, due to the use of XML for information encoding. UPnP and REST have similar performance because both of them follow the HTTP protocol.

**Listing 11:** WS Unsubscribe message

```
POST /Lamp/service.asmx HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPACTION: http://schemas.xmlsoap.org/ws/2004/08/
    eventing/
Host: 172.16.6.31:2255
Content-Length: 591

<?xml version="1.0" encoding="utf-8"?><s12:Envelope
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/
    addressing" xmlns:wse="http://schemas.xmlsoap.org/ws
    /2004/08/eventing" xmlns:s12="http://www.w3.org
    /2003/05/soap-envelope">
<s12:Header>
   <wsa:To>http://172.16.6.31:2255/Lamp/service.asmx</
       wsa:To>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/08/
       eventing/Unsubscribe</wsa:Action>
   <wsa:MessageID>uuid:3775bad6-cc95-4f23-a873-28
       aa6991e5f5</wsa:MessageID>
   <wse:Identifier>uuid:b070bf39-1c1b-429b-a809-
       ceff0825abe0</wse:Identifier>
</s12:Header>
<s12:Body>
   <wse:Unsubscribe />
</s12:Body>
</s12:Envelope>
```

The main difference, although not related to the message size, is that UPnP defines new HTTP methods (SUBSCRIBE, UNSUBSCRIBE and NOTIFY) instead of using the existing ones (PUT, POST, DELETE). These differences can be seen in Listings 12 and 13.

**Listing 12:** UPnP Unsubscribe message

```
UNSUBSCRIBE /_SwitchPower_event HTTP/1.1
HOST: 172.16.6.31:53054
SID: uuid:aa563511-44d6-4a79-a22b-40ab2108e8fb-
    SwitchPower-3
```

**Listing 13:** RESTful Unsubscribe message

```
DELETE /REST/lamps/lamp1/subscriptions/subscription0
    HTTP/1.1
Host: 172.16.6.1:8888
```

### 4.1.3 Overhead of safety related information

This section describes the results obtained with a prototype implementation of dependable RESTful messaging solutions. We specifically evaluate the impact of adding safety headers with the REST-based messages and compare parameters such as size and processing time with messages not incorporating safety mechanisms.

Figure 4.36 shows the size of different representative message types without (as shown in Listing 14) and with (as shown in Listing 15) the incorporation of safety related HTTP headers. We observe that the impact of incorporating safety headers is significant as it almost makes the message size twice. However, this approach is still lightweight and easy to implement to use ubiquitously.

**Listing 14:** Event data without safety headers

```
PUT /{AAL_id}/kitchen_camera/sensor_data HTTP/1.1
Host: 193.145.112.56
Content-Type: img/jpeg
Content-Length: 4663
```

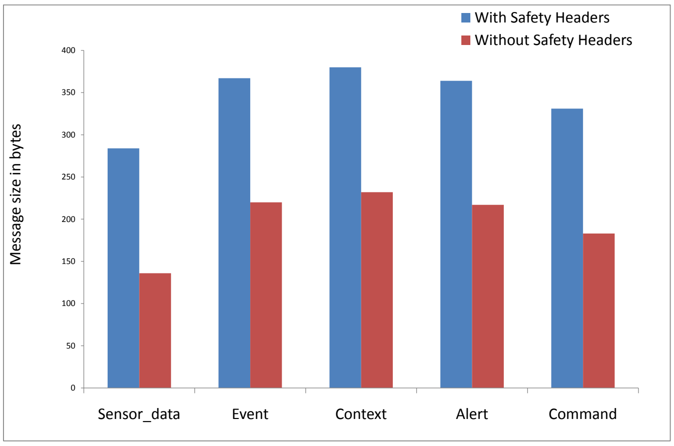


**Figure 4.36:** Message size comparison

```
 <event>
   <type>motion</type>
   <raw_data>true</raw_data>
   <location>kitchen</location>
</event>
```

Similarly, Figure 4.37 shows the impact of considering safety mechanisms in terms of performance. It shows the processing time (in ms) of a single message with or without safety headers. This is represented by the red and blue lines, respectively. The green line shows the safety verifying time, which includes the parsing and processing of safety headers and the computation of MD5 checksum to verify the integrity of the messages.

**Listing 15:** Event data with safety headers

```
PUT /{AAL_id}/kitchen_camera/sensor_data HTTP/1.1
Host: 193.145.112.56
Content-Type: img/jpeg
Content-Length: 4663
Date: Wed, 15 Apr 2014 09:22:55 GMT
From: 3e2504e0-4f89-11d3-9a0c-0305e82c3301
Content-MD5: e14a3ff5b5e67ede599cac94358e1028
Sequence-Number: 437
```

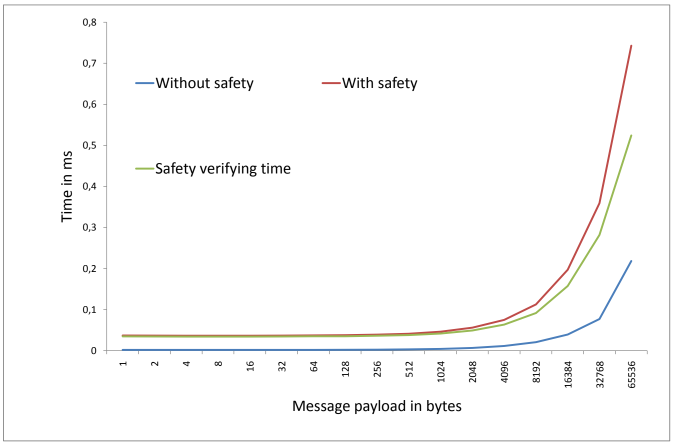


**Figure 4.37:** Processing time of safety headers

```
 <event>
   <type>motion</type>
   <raw_data>true</raw_data>
   <location>kitchen</location>
</event>
```

Figure 4.38 shows the overhead of using safety mechanisms in terms of a ratio between the time required to process safety-enabled messages with that of plain messages. These evaluations were conducted by considering different message payload sizes as shown in x-axis in Figures 4.37 and 4.38. The results shown in this figure were obtained based on the average processing time of 10000 (ten thousand) messages for each payload size.

Our experiments show that the impact of considering dependability mechanisms in message exchange in terms of performance is more significant for smaller messages (less that 128 bytes), while we observe less impact for larger messages (over 1 Kbyte).

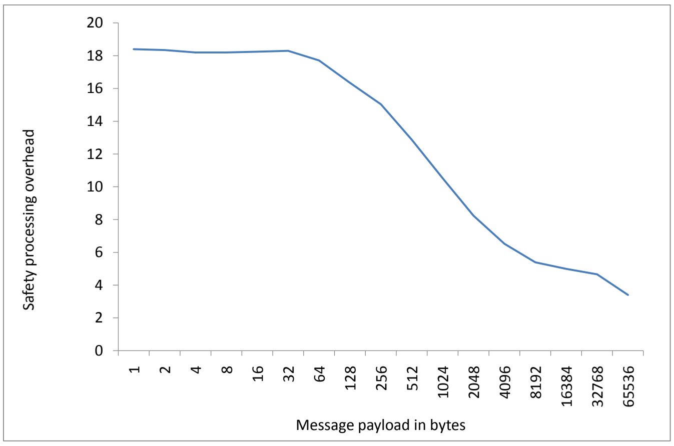


**Figure 4.38:** Overhead of safety processing time

## 4.2 FUNCTIONAL VALIDATION

This section validates the proposed solutions checking their suitability different application domains. Some prototype implementations of the collaborative decentralized environment have been developed in order to check their adequacy. The section describes the developed prototypes and the results of the implemented solutions.

### 4.2.1 Smart Home prototype

The objective of this prototype implementation is the assessment of the collaborative framework described in section 3.1, focusing on distributing the global application knowledge among the existing devices. According to the requirements described in section 3.1 for P2P architecture, the individual devices must support support discovery and eventing mechanisms. For this prototype, we used DPWS as device integration protocol.

In order to evaluate the proposed approach for collaborative *things* we implemented a proof-of-concept prototype

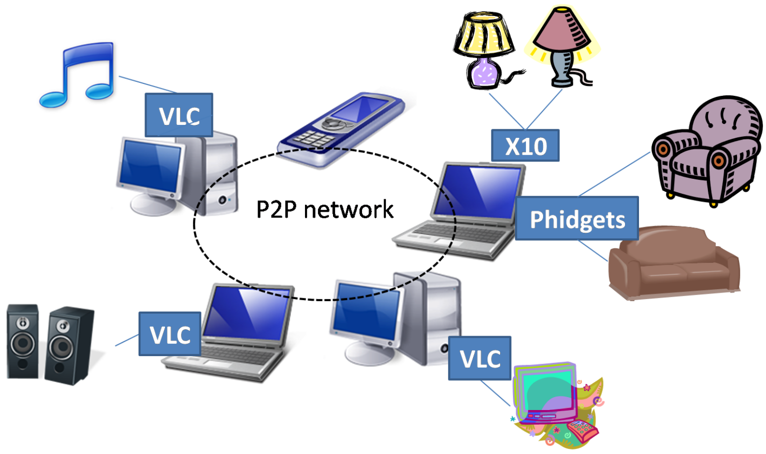


**Figure 4.39:** Smart Home environment prototype

architecture based on a set of peer devices with internal ECA rules defined to fulfill the proposed scenario in a Smart Home environment. Figure 4.39 shows the implementation environment and how the different devices are interconnected. We installed Phidget [29] pressure sensors in a couch and in an armchair and connected them using OSGi running in a laptop. We used X10 lamp modules and a PC interface for lighting control connected to the same laptop. We simulated a TV, HiFi system and Speakers services using VLC [100], a media player for various formats with a HTTP interface that allowed us to wrap a DPWS device around it. A GUI in a Smart Phone allowed us to send user actions to the devices.

We used .NET Deployment Framework and OSGi Deployment Framework tools developed in the IST Amigo Project [14] to implement the WS-Discovery and WS-Eventing protocols supporting peers, and distributed them as shown in Figure 4.39. We remark that some of the peers were implemented using OSGi ($P_{\text{Couch}}$ and $P_{\text{Armchair}}$), while the others were implemented using .NET Framework. Thus, we demonstrate the interoperability capabilities of a Web Service based solution like DPWS. For this first prototype implementation we hardcoded the ECA rules for individual peers.

Our objective was to build a flexible P2P-like home architecture with distributed behavior, so we conducted some test cases in order to demonstrate it. First, we analyzed the case of incorporating a new device in the environment. To do that, we defined a new scenario to be added to the current ones consisting on automatic TV or HiFi volume decreasing when a phone call is received. In order to realize this scenario, and still support all the others, we incorporate a new peer in the smart phone $P_{Phone}$, which launches an event whenever a phone call is received. This is all the required behavior for the new peer. We only need to add a new behavior to $P_{Speakers}$, stating that now it should also subscribe to phone events, and define the corresponding event handling actions: when receiving Phone.Call event, set the volume to low level (own action). Thus a simple update in one of the peers supports the newly added scenario.

Second, we tested the case of direct user actions in the environment. We implemented a GUI application in the user's smart phone, which is able to discover and locate all the peer services using a DPWS client ($C_{GUI}$). Thus, if user wanted to watch TV without sitting on the couch, he/she could use the phone to invoke the SwitchOn action exposed by $S_{TV}$. This invocation would directly switch the TV on, and launch the corresponding TV.On event that would be received by $C_{LightControl}$ and $C_{Speakers}$, and these two peers would act according to the defined behavior. The light level should be reduced and the Speakers source should be set to play the TV sound in this case.

Finally, we analyzed the system behavior by intentionally removing $P_{Couch}$ from the environment. This change will only affect the $P_{TV}$ peer (see Figure 3.15). By means of the SubscriptionEnd message (according to WS-Eventing protocol) and Bye message (according to WS-Discovery protocol) sent by $S_{Couch}$, the unavailability of $P_{Couch}$ was notified to $P_{TV}$. Furthermore, as a clear benefit of the behavior distribution, user could manually switch TV on, and as showed in previous test case, lights and speakers would still work as expected.

It must be remarked that the application logic distribution does not depend on the selected device integration protocol. Both REST or UPnP based solutions would have been good choices for this demonstration, because they support the discovery and eventing features required for the decentralized solution. The internal processing of the discovery and eventing messages and the implementation of the rule engine are completely isolated from the selected communication protocol.

The conclusion of this experiment is that splitting the application logic among the independent peers is a feasible solution for building collaborative environments. It is an scalable solution as demonstrated with the conducted test cases, and allows to incorporate new devices and functionalities dynamically following the ECA rules approach: the new devices must be aware of events produced by other networked devices and, according to some runtime conditions, will execute their own actions.

### 4.2.2 Ambient Media prototype

The objective of this prototype implementation is the evaluation of the suitability of the proposed collaborative framework for a different application domain. In the experiment described in section 4.2.1, the proposed approach was applied in the Smart Home domain, targeting device integration and logic distribution. This prototype implementation addresses a different smart environment domain: an application for provisioning of ambient media in the user's environment. The application components are designed as individual and distributed peers following the principles described in chapter 3.

The system must handle the different aspects related to the media selection process [40]. For example, ambient media is delivered to the user depending on their context and hence, the system needs to dynamically determine the context and provide media that are relevant therein. To set the premise of ambient media, a system may also need to customize the physical environment, for example, by dimming the lighting level or by lowering the volume. Besides, users' need for media services changes over time and space that requires mechanisms to continually update their preferences based on their mobility in the environment. In this experiment, we propose an ambient media service provisioning framework that incorporates the above requirements while keeping the user at the center of the media selection loop. To demonstrate the usefulness of this framework, we show experimental results by considering real-life scenario in a smart home environment and in a mobile user's context.

#### 4.2.2.1 *Problem Description*

In the context of ambient intelligence [26], user's environment is technologically augmented with myriad sensors and networked devices in order to provide support to the users. In such an environment, providing ambient media to the users based on their context is a challenging issue. This requires addressing several key issues, which are context determination, user's preference management, control of smart artifacts, and managing users mobility. Besides, the development of a framework that can incorporate these aspects in a coherent manner is not a trivial issue. In the following, we highlight the difficulties that appear while addressing the above issues.

- User's context plays an important role while delivering ambient media in the environment. Several sensory data need to be fused to determine the context. However, context determination is not always straight forward due to the imprecision of sensing and media processing tasks. The dynamic derivation of context and the granularity of context parameters also brings challenge in the overall task.

- User's media preference changes over time, which need to be managed in order to recommend users with relevant and appropriate media at a particular context. The dynamic update of preference need to consider user's context and interaction history to reflect user's contextual preference, which is later used to estimate the gain of available media services. However, how to incorporate interaction history to update the changing preference and how to estimate gain remains to be an important issue.

- Providing ambient media may also require to customize the environment in order to provide better quality of experience. Environment customization relates to the control of several household artifacts, such as changing the lighting intensity or lowering the volume of an audio device. The heterogeneity of devices and artifacts as well as their interconnections need to be considered in this respect.
- It is usual that the mobility of the users change. However, they should be provided with ambient media wherever they are as per their need. Therefore, a mechanism is needed not only to obtain their updated preferences but also to make it available in mobile locations to personalize their media selection.
- The development of the framework that incorporates the above issues also poses challenge. This is mostly related to practical issues such as the use of particular technology, the communication among several components, the availability and access of the media, the interconnection of the devices to name a few.

#### 4.2.2.2 *Proposed Framework*

In this section, we provide an overview of the ambient media provisioning loop to clarify the design of the proposed framework, Framework for Ambient Media (FAMe). Figure 4.40 shows this loop [39]. As per this figure, it is clear that ambient media is centered around the users. The users interact in the environment while context is identified by the context identification process using several context sensors. Based on the interaction history in a particular context, user's ambient media preference gets updated that is later used along with the reputation of media services to estimate the gain of the available media services. The service selection process afterwords uses this gain to select and provide media to the user. The realization of the whole media provisioning process requires the development of several components, which should work together in a logical sequence.

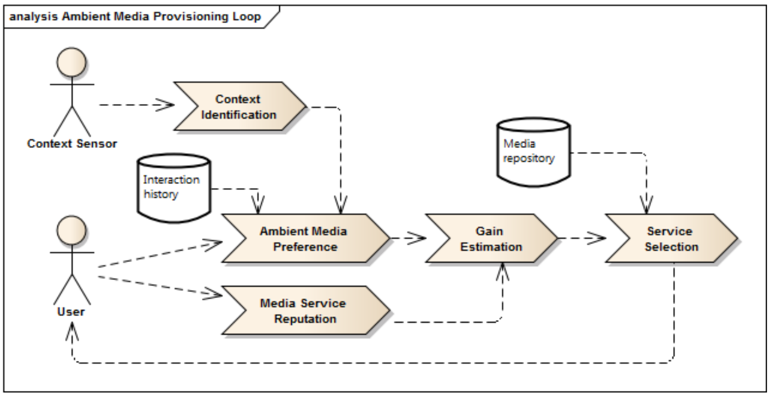


**Figure 4.40:** Schematic view of the ambient media provisioning Loop

The proposed framework dynamically maintains a set of media preferences for the user based on different contexts. We call this as Ambient Media Preference (AMP)[39], which contains the scores of metadata-related attributes of different types of media services.

We now draw the high-level architecture of the proposed framework in Figure 4.41, which highlights its core components. In this architecture we show how it incorporates the several aspects earlier mentioned:

- The users context is handled by the Context Identifier module.
- Ambient media preference update and gain estimation is instrumented by the Ambient Media Selector module along with the Interaction Manager module, the Media Source Manager and the Gain Manager module
- The control of environment artifacts is taken care by the Ambient Controller module in cooperation with other control module such as smart space automation and media renderer
- User's mobility management is addressed by using the users mobile device as the interface to access media and to store ambient media preference

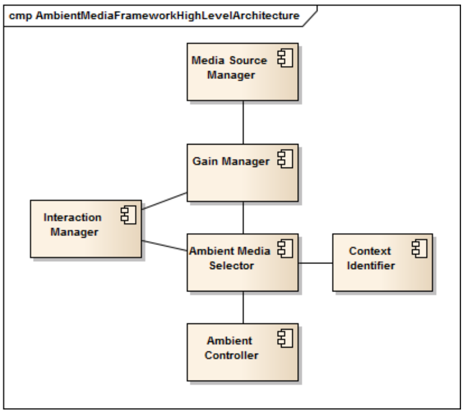


**Figure 4.41:** High level architecture of the proposed ambient media provisioning framework.

In this section we describe how the different aspects described above are integrated in FAMe. Figure 6 shows the overall component model of FAMe. Note, this is a more detail version of the high-level architecture presented in Figure 4.41. In Figure 4.42, we have explicitly shown the corresponding stereotype with the links that follow between any two components. Note that in this figure, the Ambient Media Selector component acts as a coordinating components to establish connections among other components to perform ambient media selection tasks. The Media Source Manager connects to different external database for extracting the media descriptor. The Context Identifier is connected to several sensing devices based on which it identifies user's current context. The Ambient Controller component uses the support of Home Automation component to control the environment artifacts. It also uses the Media Renderer component to select the appropriate rendering device and to control the volume of the audio output. The Gain Manager component collaborates with the Interaction Manager, Media Source Manager and AMP Manager to dynamically update the AMP scores based on the interaction history. The AMP Manager handles the access of the AMP scores to/from the AMP score repository. In our framework we propose to keep the AMP Manager and AMP score repository in the user's portable device.

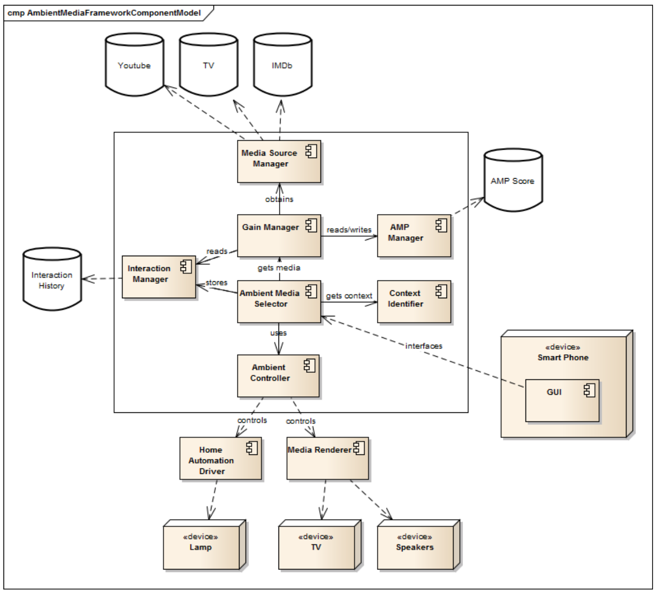


**Figure 4.42:** Component model of the proposed ambient media provisioning framework.

In FAMe, a user gets support from the proposed framework to select the appropriate and relevant ambient media based on his/her current context. The media selection can occur in two different modes, which are automatic support mode and semi-automatic support mode.

1. Automatic support

   In this mode, a user merely indicates the system that he/she is willing to experience ambient media. The system automatically takes care of the other issues, which includes the determination of context, suggesting the media that has the maximum gain, and finally rendering the media in the appropriate rendering device. In light of the overall component architecture (Figure 4.42), we provide a sequence diagram in Figure 4.43 to show the sequence of interactions that take place among the components in the automatic support mode.

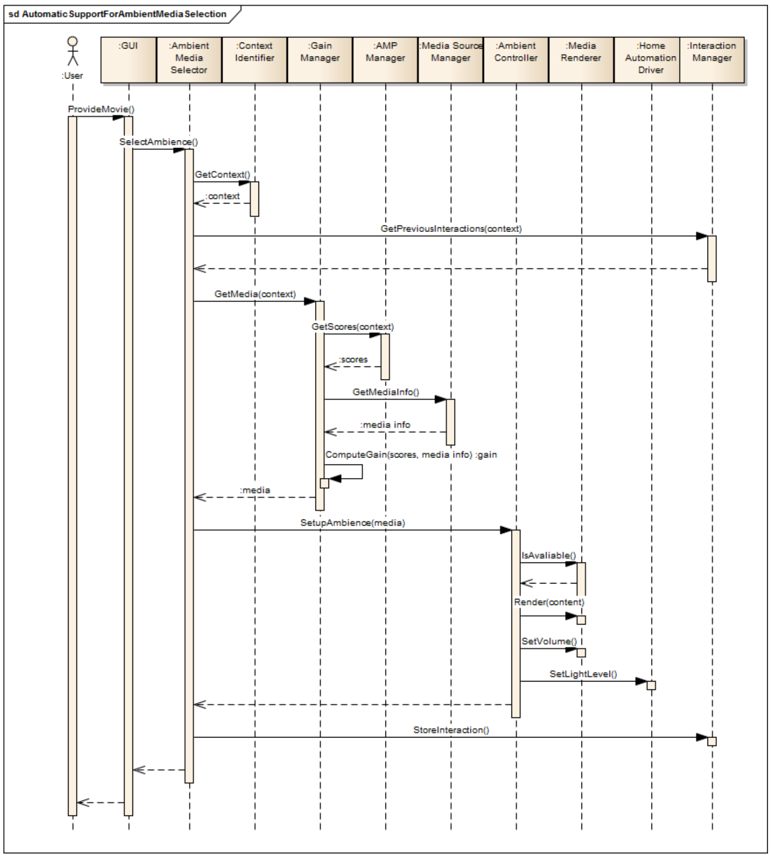


**Figure 4.43:** Sequence of interactions that take place among the different components in automatic support mode for media provision.

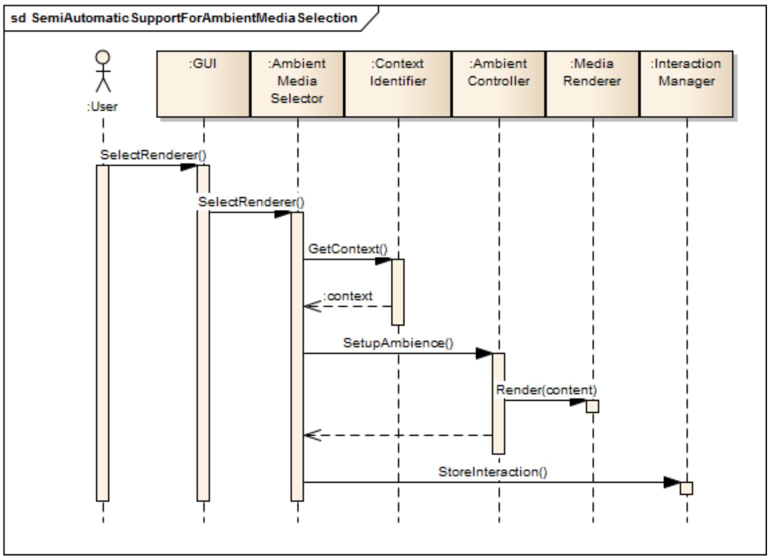


**Figure 4.44:** Sequence of interactions that take place in semi-automatic support mode provision. Note, in this figure we only show a case when the user changes the renderer selection that has been proposed by the system.

2. Semi-automatic support

   A user in this mode may choose to change the automatic media selection by changing the recommended media and the different other settings such as a rendering device. Basically, in this mode also the system recommends appropriate media and ambiance to the user as soon as the user asks for it. However, this option gives the user facility to override the media selection that is made by the system. Figure 4.44 shows the sequence of steps that occur in an example case of the change of renderer by the user. Similarly, the change of other settings takes place.

#### 4.2.2.3 *P2P based communication*

The connections among the components have been implemented using P2P paradigm. The service components offer a

Web Service port as exposed interface of internal business logic, and the client components use Web Service clients as proxies of the desired service. For the sake of simplicity, we have only shown the business logic part in some of the components in this figure. Note, in our design, the AMP Manager and Ambient Media Selector GUI resides in the smart phone.

#### 4.2.2.4 *Framework Prototype Implementation*

This section shows the layout of the experimental smart environment and the different devices that are in that environment. In this environment, all the rooms are equipped with X10 lamps allowing user or system control of the light level. The environment is also equipped with Ubisense [90] indoor location system, which provides continuous 3D positioning of any object or user equipped with an active tag. Table 4.17 lists the different devices that are in the environment, which we use to conduct our experiment. Besides the devices listed in Table 4.17, all the networked devices are connected to a router using Ethernet (Media Center + Kitchen PC) and alternatively using WiFi connection (Laptop, SmartPhone and Ultra Mobile PC).

- Context Identification Our context identification service primarily depends on Ubisense [90] real-time location system. Using Ultra Wide Band (UWB) radio frequency technology, Ubisense location system proportionates 3D location of active tags (connected to people and/or device) in indoor environments. The Ubisense platform offers a .NET based Software Development Kit (SDK), allowing us to develop a ASP.NET Web Service wrapper to provide real-time location information of users and mobile objects at home. Therefore, using this technology we obtain the values of different context attributes such as who is around, their location and time.

- AMP Update and Gain Management Service

  We maintain user's interaction history to dynamically update user's ambient media preference. We use C# code to write a method for this purpose and later wrap it using in a ASP.NET web service so that this can be utilized easily whenever AMP update is needed.

We develop another web service that collects rating data from the web (e.g. IMDB movie database) and uses this rating (if available) with with updated AMP scores to compute gain in a media service. Note that the estimated gain value is used by the Ambient Media Selector to recommend the user a media that has the highest gain.

- Media Server

  We adopted the technique described in [11] to develop our media server. It is an OSGi based Digital Video Broadcasting (DVB)-S and DVB-C Tuner for TV content streaming using Real-time Transport Protocol (RTP)-Real Time Streaming Protocol (RTSP) protocol. We used the MediaCenter in our environment to store movies and songs. For content metadata acquisition, we consulted Internet Movie Database (IMDb) and TheTVDB.com through a web service that is built on top of the API provided by the respective parties.

- Media Renderer The Media Renderer Service in our design has been implemented using VLC [100], which is a highly portable multimedia player supporting various streaming protocols and audio and video formats. VLC integrates a little HTTP server that can be used for a HTTP remote control interface. We developed a wrapper that controls the VLC player using VLC's HTTP interface.

- Home Automation Service We use X10 lamp modules and the ActiveHome CM11A Computer Interface to develop a ASP.NET Web Service that facilitates the lighting control in each room. Thus, the lighting level of each room can be set individually using standard Web Service protocols.

- Media Provisioning Service We developed a ASP.NET Web Service for the Media Provisioning Service, running on the Kitchen PC, and acting as the entry point of

| Location | Devices | Description |
|---|---|---|
| Living room | Media center | Fujitsu-Siemens Scaleo E, running Windows XP Media Center Edition |
| | Speakers | Ambient sound |
| | 40" TV screen | Media renderer |
| Bedroom | Laptop | |
| Kitchen | 19" Monitor | Media renderer |
| | Speakers | Ambient sound |
| | PC | X10 home automation interface is connected to this PC |
| Mobile devices | Ultra Mobile PC | Used as a mobile renderer |
| | Smart Phone | User interface |

**Table 4.17:** Devices in the experimental Ambient Media prototype.

ambient media selection at home. A Web Service client application with GUI in the user's Smart Phone has been developed that access the proposed framework and uses the media provisioning service to ask for ambient media.

### 4.2.3 Health management prototype

The objective of this prototype implementation is the validation of the RESTful discovery and eventing specifications described in sections 3.2 and 3.3. To do so, we analyze its suitability for e-Health applications in AAL environment. In order to investigate the suitability of the proposed approach in AAL, this section describes a prototype pervasive application for healthcare support in a home environment that has been implemented using the RESTful discovery and eventing specifications, proposed in this dissertation. Nowadays several health monitoring sensors (*things*) are available in the market with heterogeneous networking capabilities, mainly Bluetooth and USB connections.


**Figure 4.45:** Healthcare sensors used in the AAL environment.

We took a set of representative sensors and embedded them in an AAL environment in order to capture different physiological parameters like blood pressure, $O_2$ saturation, weight, temperature, glycemia, cholesterol, electrocardiogram (ECG) and spirometry, as shown in Figure 4.45.

This prototype realizes a monitoring system that seamlessly captures and stores data from a set of heterogeneous health sensors as mentioned before, displaying elaborated health information to users by means of a front-end application, running in a SmartTV and a tablet. The prototype demonstrates a distributed architectural deployment of the healthcare application as shown in Figure 4.46. The sensors layer is composed of a number of devices with heterogeneous communication capabilities and device-dependent protocols such as USB and Bluetooth serial communication profiles.

It is possible that a sensor device implements the proposed mechanism in order to support dynamic discovery and eventing policies in a RESTful way, however none of the sensors used in this prototype had those capabilities. Without this support, the devices need to be integrated using a different mechanism, such

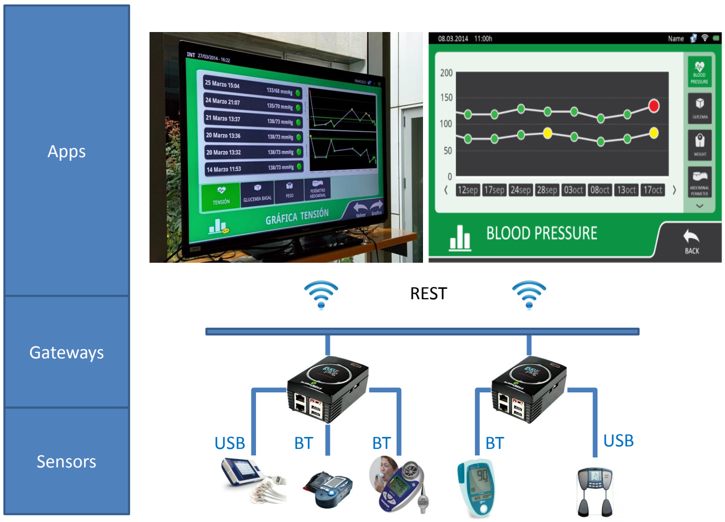


Figure 4.46: Application architecture

| Devices | Model |
|---|---|
| Blood pressure | Taidoc TD-3132 |
| $O_2$ Saturation | Digi$O_2$ HemOxi POM-101 |
| Weight | Tanita BC-590BT |
| Glycemia | Taidoc TD-4255 |
| ECG | Taidoc TD-2202 |
| Spirometry | Vitalograph copd-6 |
| Temperature | Taidoc TD-1261A |
| Plug computer | Globalscale GuruPlug Server |
| SmartTV | Sony NSZ-GS7 |
| Tablet PC | Archos 80 G9 |

Table 4.18: Integrated devices for healthcare application.

as the one we adopted with the help of several plug computers running Debian/Linux operating system.

We then built a Java prototype implementation of the RESTful service discovery and eventing model which acts as a gateway from the device proprietary protocols to a Wi-Fi network, exposing all the health devices as REST resources that can be discovered and subscribed by any potential application. These applications benefit from the discovery and eventing capabilities offered by the proposed approach.

The Application layer in the figure is developed as a Java based user interface, that tracks and monitors data coming from the available sensors, running in Android home devices (SmartTV and tablet). The application is built on top of a homogeneous middleware, a prototype of the proposed approach, which offers discovery and eventing mechanisms, thereby isolating the application from the variability of underlying device protocols.

The specific set of hardware components used for this prototype implementation is listed in Table 4.18.

This experiment implemented a RESTful discovery and eventing specification, which is a lightweight and efficient approach compared to existing service discovery and event management solutions.

This prototype implements the specification of the RESTful resource tree structure, resource tree instantiation for sample event source and event listener, and interactions between resources of event source and event listener for both discovery and eventing mechanisms.

We evaluated the proposed approach with existing ones as per specification and found that the RESTful approach needs minimal message size for service discovery and eventing, which is very promising. We implemented the proposed technique in the context of an assisted living healthcare scenario for connecting heterogeneous sensor and actuator devices in order to obtain real-time health data from the inhabitants for better health monitoring.

The proposed approach described here would be a suitable solution for many pervasive sensor-rich environments for dynamic service provisioning. Therefore, in the future we plan to demonstrate the applicability of our approach in various pervasive applications domain, and to study the requirements for safety and security that come together with the use of health related data.

# 5 CONCLUSIONS

This chapter concludes the thesis, summarizing the obtained contributions in section 5.1. Section 5.2 shows the list of publications resulted from this research work. Finally, section 5.3 suggests and points out some future work in this area.

## 5.1 SUMMARY OF CONTRIBUTIONS

This thesis has proposed a reference framework suitable for IoT that considers the following aspects:

- Decentralized collaborative architecture built upon distributed *things*.
  As a result of **Research Activity 1**, it has discussed a P2P-like scheme for a collaborative IoT architecture and analyzed the possibility of distributing the required application workflow logic to individual peers. In this approach, each peer represents a *thing* and acts autonomously based on the application logic it has for its own interaction.

  The proposed approach not only provides the flexibility of adding or removing new or existing *thing* to/from the *Internet*, but it also ensures scalability and removes the burden from the central entity usually encountered in a traditional centralized server or gateway based solutions.

- RESTful discovery and eventing mechanisms for the *things*.
  As a result of **Research Activity 2**, a lightweight middleware approach has been proposed, based on REST principles for enabling dynamic discovery and event management of networked *things*. More specifically, a mechanism to dynamically discover networked *things* and

exchange notifications among them has been detailed, following the REST principles. Such a middleware is suitable for service provisioning in IoT scenarios.

- Representation of the *things* according to IEC 61850 based information model.
  As a result of **Research Activity 3**, a RESTful mapping for IEC 61850 services has been specified in this research, enabling the adoption of the principles and reference architecture proposed by this standard for IoT applications, having the *Internet* as the scope of the networked *things*.

- Dependable RESTful communication protocols for the *Internet*.
  As a result of **Research Activity 4**, this work has analyzed the variability of messaging requirements for dependable smart environments, as an specialization of IoT, and has proposed a dependability-enabled RESTful communication pathway for alert-response management, incorporating techniques and measures to detect communication errors and faults.

  It has also proposed a framework of alert-response messaging suitable for a dependable IoT environment, showing the variability of messages being exchanged in such an environment and identifying the need to incorporate dependable messaging for communication among the different components of the framework.

  It adopts the RESTful principles to specify the different message formats for safety-enabled communication in an AAL environment and evaluated the impact of safety mechanisms on the overall message exchange. This approach can be used in many other control and monitoring applications with functional safety requirements.

## 5.2 DISSEMINATION

The following peer-reviewed journal and conference papers were published during the course of this thesis:

### 5.2.1 International Journals

- *From Sensing to Alerting: a Pathway of RESTful Messaging in Ambient Assisted Living.*
  Hossain MA, Parra J, Rahman SMM, Alamri A, Ullah S, Mouftah H. .
  **(Accepted) IEEE Wireless Communications Magazine. (IF 5.417)**

- *A Framework for a Context-Aware Elderly Entertainment Support System.*
  Hossain MA, Alamri A, Almogren AS, Hossain SK, Parra J.
  **(2014) Sensors, 14(6), 10538-10561. (IF 2.25)**

- *RESTful Discovery and Eventing for Service Provisioning in Assisted Living Environments.*
  Parra J, Hossain MA, Uribarren A, Jacob E.
  **(2014) Sensors, 14(5), 9227-9246. (IF 2.25)**

- *A framework for human-centered provisioning of ambient media services.*
  Hossain MA, Parra J, Atrey PK, El Saddik A
  **(2009) Multimedia Tools and Applications, 44(3), 407-431. (IF 1.35)**

- *Flexible smart home architecture using device profile for web services: A peer-to-peer approach.*
  Parra J, Hossain MA, Uribarren A, Jacob E, El Saddik A.
  **(2009) International Journal of Smart Home, 3(2), 39-56.**

### 5.2.2 International Conferences

- *Safety-enabled restful messaging in Ambient-Assisted Living.*
  Hossain MA, Parra J, Alamri A
  **In Multimedia and Expo Workshops (ICMEW), 2015 IEEE International Conference on (pp. 1-6). IEEE.**

- *Context-aware elderly entertainment support system in assisted living environment.*
  Hossain MA, Alamri A, Parra J
  **In Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on (pp. 1-6). IEEE.**

- *Bridging the Gap between Services and Context in Ubiquitous Computing Environments Using an Effect-and Condition-Based Model.*
  Urbieta A, Azketa E, Gomez I, Parra J, Arana N.
  **In 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008 (pp. 149-158). Springer Berlin Heidelberg.**

- *A middleware platform for application configuration, adaptation and interoperability.*
  Uribarren A, Parra J, Iglesias R, Uribe JP, López-de-Ipina D
  **In Self-Adaptive and Self-Organizing Systems Workshops, 2008. Second IEEE International Conference on (pp. 162-167). IEEE.**

- *A survey of dynamic service composition approaches for ambient systems.*
  Urbieta A, Barrutieta G, Parra J, Uribarren A
  **In Proceedings of the 2008 Ambi-Sys workshop on Software Organisation and MonIToring of ambient systems (p. 1). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).**

- *Middleware for distributed services and mobile applications.* Uribarren A, Parra J, Uribe JP, Zamalloa M, Makibar K **In Proceedings of the first international**

**conference on Integrated internet ad hoc and sensor networks 2006. ACM.**

- *The Amigo interoperable middleware for the networked home environment.*
  Sacchetti D, Bromberg Y, Georgantas N., Issarny V, Parra J, Poortinga R
  **In 6th International Middleware Conference 2005, Workshops Proceedings, Grenoble, France.**

## 5.3 FUTURE WORK

Current devices, applications and systems based on the IoT paradigm do not have the same level of reactivity or context awareness described in the areas proposed by Lassila [52], Weiser [105] or the ISTAG [26] because they are either simple device systems which behave in a basic way like opening doors when a person is detected around, switching on the light when a person is coming, redirecting a call to the closest telephone, etc.; or they are just centralized systems with a central computer and devices located around it just receive working commands.

In order to achieve a more intelligent, advanced and impromptu behavior with the ability to reason and learn independently, intelligent environment *things* must have some degree of intelligence and must be light, small and consume low energy, so they can be embedded in every day gadgets. This kind of intelligent environments should be built only in embedded systems, instead of in systems requiring big devices like PCs, which create artificial and unreal scenarios, very far away from the idea of ubiquitous computing.

A happy solution should be investigated, based on a light (small and with low energy consumption) system but still intelligent enough to be able to reason over user context, perceived inputs and the desired objectives, and capable of communicating and coordinating with other *things* located in the *Internet* scale environment to perform more complex tasks.

# BIBLIOGRAPHY

[1] E. Aarts and R. Wichert. *Ambient intelligence*. Springer, 2009.

[2] E. AbuKhousa, N. Mohamed, and J. Al-Jaroodi. e-health cloud: opportunities and challenges. *Future Internet*, 4 (3):621–645, 2012.

[3] M. Aiello. The role of web services at home. In *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, AICT/ICIW'06*, volume 2006, page 164, 2006.

[4] O. Alliance. *Osgi service platform*. IOS Press, Inc., 2003.

[5] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[6] J. C. Augusto and C. D. Nugent. The use of temporal reasoning and management of complex events in smart homes. In *ECAI*, volume 16, page 778, 2004.

[7] J. Beatty. Web services dynamic discovery (ws-discovery), 2004.

[8] U. Bellur and N. C. Narendra. Towards service orientation in pervasive computing systems. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, pages 289–295. IEEE, 2005.

[9] F. Belqasmi, R. Glitho, and C. Fu. Restful web services for service provisioning in next-generation networks: a survey. *Communications Magazine, IEEE*, 49(12):66–73, 2011.

[10] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[11] J. Bilbao and I. Armendariz. Convergence in digital home communications to redistribute iptv and high definition contents. In *2007 4th IEEE Consumer Communications and Networking Conference*, 2007.

[12] A. Bottaro, J. Bourcier, C. Escoffier, and P. Lalanda. Context-aware service composition in a home control gateway. In *Pervasive Services, IEEE International Conference on*, pages 223–231. IEEE, 2007.

[13] A. Bottaro, A. Gérodolle, and P. Lalanda. Pervasive service composition in the home network. In *Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on*, pages 596–603. IEEE, 2007.

[14] A. Bottaro, E. Simon, S. Seyvoz, and A. Gérodolle. Dynamic web services on a home service platform. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pages 378–385. IEEE, 2008.

[15] J. Bourcier, A. Chazalet, M. Desertot, and C. Marin. A dynamic-soa home control gateway. In *null*, pages 463–470. IEEE, 2006.

[16] D. Box, L. F. Cabrera, C. Critchley, F. Curbera, D. Ferguson, A. Geller, S. Graham, D. Hull, G. Kakivaya, and A. Lewis. Web services eventing (ws-eventing). *W3C member submission*, 15:355, 2006.

[17] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6): 599–616, 2009.

[18] K. Carey, D. Lewis, S. Higel, and V. Wade. Adaptive composite service plans for ubiquitous computing. In *2nd International workshop on managing ubiquitous communications and services (MUCS 2004)*, 2004.

[19] D. Chakraborty. Service discovery and composition in pervasive environments. *Ph. D. Thesis*, 2004.

[20] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha. Service composition for mobile environments. *Mobile Networks and Applications*, 10(4):435–451, 2005.

[21] S. Cotofana, S. Wong, S. Vassiliadis, and S. Stamatis. Embedded processors: Characteristics and trends. In *Proceedings of the 2001 ASCI Conference*, pages 17–24, 2001.

[22] T. Cottenier and T. Elrad. Adaptive embedded services for pervasive computing. In *Workshop on Building Software for Pervasive Computing-ACM SIGPLAN conf. on Object-Oriented Programming, Systems, Languages, and Applications*. Citeseer, 2005.

[23] V. G. Da Silva, C. E. Cirilo, A. F. do Prado, and W. L. de Souza. An approach to dynamic discovery of context-sensitive web services. In *Proceedings of the Eighth International Conference on Internet and Web Applications and Services, Rome, Italy*, pages 83–89, 2013.

[24] C. N. Doukas and I. Maglogiannis. Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components. *Information Technology in Biomedicine, IEEE Transactions on*, 15(2):277–289, 2011.

[25] W. Drytkiewicz, I. Radusch, S. Arbanowski, and R. Popescu-Zeletin. prest: a rest-based protocol for pervasive systems. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 340–348. IEEE, 2004.

[26] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman. *Scenarios for ambient intelligence in 2010*. Office for official publications of the European Communities, 2001.

[27] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2):114–131, 2003.

[28] R. T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.

[29] C. Fitchett and S. Greenberg. The phidget architecture: Rapid development of physical user interfaces. In *Proc. Workshop Application Models and Programming Tools for Ubiquitous Computing*, 2001.

[30] A. Forkan, I. Khalil, and Z. Tari. Cocamaal: A cloud-oriented context-aware middleware in ambient assisted living. *Future Generation Computer Systems*, 35:114–127, 2014.

[31] U. Forum. Upnp forum homepage, 2012.

[32] D. Griffin and D. Pesch. A survey on web services in telecommunications. *Communications Magazine, IEEE*, 45(7):28–35, 2007.

[33] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[34] D. Guinard, V. Trifa, and E. Wilde. A resource oriented architecture for the web of things. In *Internet of Things (IOT), 2010*, pages 1–8. IEEE, 2010.

[35] D. Guinard, I. Ion, and S. Mayer. In search of an internet of things service architecture: Rest or ws-*? a developers' perspective. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 326–337. Springer, 2012.

[36] V. C. Güngör, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke. Smart grid technologies: communication technologies and standards. *Industrial informatics, IEEE transactions on*, 7(4):529–539, 2011.

[37] S. Helal, W. Mann, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: A programmable pervasive space. *Computer*, 38(3):50–60, 2005.

[38] C. Hesselman, A. Tokmakoff, P. Pawar, and S. Iacob. Discovery and composition of services for context-aware systems. In *Smart Sensing and Context*, pages 67–81. Springer, 2006.

[39] M. A. Hossain, P. K. Atrey, and A. El Saddik. Dynamic gain estimation in ambient media services. In *Proceedings of the 1st ACM international workshop on Semantic ambient media experiences*, pages 49–56. ACM, 2008.

[40] M. A. Hossain, J. Parra, P. K. Atrey, and A. El Saddik. A framework for human-centered provisioning of ambient media services. *Multimedia Tools and Applications*, 44 (3):407–431, 2009.

[41] IEC. *IEC 61850 Communication networks and systems in substations - Part 8-1: Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*. International Electrotechnical Commission, 2004.

[42] IEC. *IEC 61400 Wind turbines - Part 25-1: Communications for monitoring and control of wind power plants - Overall description of principles and models*. International Electrotechnical Commission, 2006.

[43] IEC. *IEC 61850 Communication networks and systems for power utility automation - Part 7-410: Hydroelectric power plants*. International Electrotechnical Commission, 2007.

[44] IEC. *IEC 61784-3-3: Industrial communication networks - Profiles - Part 3-3: Functional safety fieldbuses*. International Electrotechnical Commission, 2007.

[45] IEC. *IEC 61400 Wind turbines - Part 25-4: Communications for monitoring and control of wind power*

*plants - Mapping to communication profile*. International Electrotechnical Commission, 2008.

[46] IEC. *IEC 61850 Communication networks and systems for power utility automation - Part 7-420: Basic communication structure - Distributed energy resources logical nodes*. International Electrotechnical Commission, 2009.

[47] V. Issarny, D. Sacchetti, F. Tartanoglu, F. Sailhan, R. Chibout, N. Levy, and A. Talamona. Developing ambient intelligence systems: A solution based on web services. *Automated Software Engineering*, 12(1):101–137, 2005.

[48] F. Jammes, A. Mensch, and H. Smit. Service-oriented device communications using the devices profile for web services. In *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*, pages 1–8. ACM, 2005.

[49] A. Jestratjew and A. Kwiecień. Performance of http protocol in networked control systems. *Industrial Informatics, IEEE Transactions on*, 9(1):271–276, 2013.

[50] G. Kaefer, R. Schmid, G. Prochart, and R. Weiss. Framework for dynamic resource-constrained service composition for mobile ad hoc networks. In *UBICOMP, workshop on system support for ubiquitous computing*, 2006.

[51] D. Kalofonos, F. Reynolds, and T. Nrc. Task-driven end-user programming of smart spaces using mobile devices. *Published in Nokia Research Center technical report (NRC-TR-2006-001)*, 2006.

[52] O. Lassila and M. Adler. 12 semantic gadgets: Ubiquitous computing meets the semantic web. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, page 363, 2005.

[53] C. Lee, S. Ko, S. Lee, W. Lee, and S. Helal. Context-aware service composition for mobile network environments. In *Ubiquitous Intelligence and Computing*, pages 941–952. Springer, 2007.

[54] S. Y. Lee, J. Y. Lee, and B. I. Lee. Service composition techniques using data mining for ubiquitous computing environments. *International Journal of Computer Science and Network Security*, 6(9):110–117, 2006.

[55] J. Lima, C. Lima, V. Gomes, J. F. Martins, J. Barata, L. Ribeiro, and G. Cândido. Dpws as specific communication service mapping for iec 61850. In *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, pages 193–198. IEEE, 2011.

[56] Z. Maamar, S. K. Mostefaoui, and H. Yahyaoui. Toward an agent-based and context-oriented approach for web services composition. *Knowledge and Data Engineering, IEEE Transactions on*, 17(5):686–697, 2005.

[57] R. Mackiewicz. Overview of iec 61850 and benefits. In *Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES*, pages 623–630. IEEE, 2006.

[58] R. Masuoka, B. Parsia, and Y. Labrou. Task computing–the semantic web meets pervasive computing. In *The Semantic Web-ISWC 2003*, pages 866–881. Springer, 2003.

[59] S. A. McIlraith, T. C. Son, and H. Zeng. Semantic web services. *IEEE intelligent systems*, 2:46–53, 2001.

[60] A. Mingkhwan, P. Fergus, O. Abuelmaátti, M. Merabti, B. Askwith, and M. Hanneghan. Dynamic service composition in home appliance networks. *Multimedia Tools and Applications*, 29(3):257–284, 2006.

[61] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[62] S. B. Mokhtar, D. Fournier, N. Georgantas, and V. Issarny. Context-aware service composition in pervasive computing environments. In *Rapid Integration of Software Engineering Techniques*, pages 129–144. Springer, 2006.

[63] S. B. Mokhtar, N. Georgantas, and V. Issarny. Cocoa: Conversation-based service composition in pervasive computing environments with qos support. *Journal of Systems and Software*, 80(12):1941–1955, 2007.

[64] S. K. Mostéfaoui, A. Tafat-Bouzid, and B. Hirsbrunner. Using context information for service discovery and composition. In *IiWAS*, 2003.

[65] M. Nakamura, H. Igaki, H. Tamada, and K.-i. Matsumoto. Implementing integrated services of networked home appliances using service oriented architecture. In *Proceedings of the 2nd international conference on Service oriented computing*, pages 269–278. ACM, 2004.

[66] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K.-i. Matsumoto. Adapting legacy home appliances to home network systems usingweb services. In *Web Services, 2006. ICWS'06. International Conference on*, pages 849–858. IEEE, 2006.

[67] Q. Ni. Service composition in ontology enabled service oriented architecture for pervasive computing. In *Workshop on Ubiquitous Computing and e-Research*, pages 242–257. Citeseer, 2005.

[68] OASIS. Devices profile for web services, 2009.

[69] F. Palumbo, J. Ullberg, A. Štimec, F. Furfari, L. Karlsson, and S. Coradeschi. Sensor network infrastructure for a home care monitoring system. *Sensors*, 14(3):3833–3860, 2014.

[70] J. Parra, M. A. Hossain, A. Uribarren, E. Jacob, and A. El Saddik. Flexible smart home architecture using device profile for web services: a peer-to-peer approach. *International Journal of Smart Home*, 3(2):39–50, 2009.

[71] C. Pautasso, O. Zimmermann, and F. Leymann. Restful web services vs. big web services: making the right architectural decision. In *Proceedings of the 17th international conference on World Wide Web*, pages 805–814. ACM, 2008.

[72] A. B. Pedersen, E. B. Hauksson, P. B. Andersen, B. Poulsen, C. Træholt, and D. Gantenbein. Facilitating a generic communication interface to distributed energy resources: Mapping iec 61850 to restful services. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 61–66. IEEE, 2010.

[73] T. Perumal, A. R. Ramli, C. Y. Leong, S. Mansor, and K. Samsudin. Interoperability for smart home environment using web services. *International Journal of Smart Home*, 2(4):1–16, 2008.

[74] H. Pourreza and P. Graham. On the fly service composition for local interaction environments. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 6 pp.–399. IEEE, 2006.

[75] A. Qasem, J. Heflin, and H. Muñoz-Avila. Efficient source discovery and service composition for ubiquitous computing environments. In *Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications (ISWC)*, 2004.

[76] Y. Qiao, K. Zhong, H. Wang, and X. Li. Developing event-condition-action rules in real-time active database. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 511–516. ACM, 2007.

[77] L. Qiu, Z. Shi, and F. Lin. Context optimization of ai planning for services composition. In *e-Business Engineering, 2006. ICEBE'06. IEEE International Conference on*, pages 610–617. IEEE, 2006.

[78] A. Ranganathan and R. H. Campbell. Autonomic pervasive computing based on planning. In *Autonomic Computing, 2004. Proceedings. International Conference on*, pages 80–87. IEEE, 2004.

[79] A. Ranganathan and S. McFaddin. Using workflows to coordinate web services in pervasive computing environments. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 288–295. IEEE, 2004.

[80] L. Richardson and S. Ruby. *RESTful web services*. " O'Reilly Media, Inc.", 2008.

[81] G. N. Rodrigues, V. Alves, R. Silveira, and L. A. Laranjeira. Dependability analysis in the ambient assisted living domain: An exploratory case study. *Journal of Systems and Software*, 85(1):112–131, 2012.

[82] D. Romero, G. Hermosillo, A. Taherkordi, R. Nzekwa, R. Rouvoy, and F. Eliassen. Restful integration of heterogeneous devices in pervasive environments. In *Distributed Applications and Interoperable Systems*, pages 1–14. Springer, 2010.

[83] B. d. Ruyter and E. Pelgrim. Ambient assisted-living research in carelab. *Interactions*, 14(4):30–33, 2007.

[84] B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5): 22–32, 1994.

[85] M. Sheshagiri, N. Sadeh, and F. Gandon. Using semantic web services for context-aware mobile applications. In *MobiSys 2004 Workshop on Context Awareness*, 2004.

[86] E. Sirin and B. Parsia. Planning for semantic web services. In *Semantic Web Services Workshop at 3rd International Semantic Web Conference*, pages 33–40, 2004.

[87] H. Song, D. Kim, K. Lee, and J. Sung. Upnp-based sensor network management architecture. In *Proc.*

*International Conference on Mobile Computing and Ubiquitous Networking*, 2005.

[88] Z. Song, Y. Labrou, and R. Masuoka. Dynamic service discovery and management in task computing. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 310–318. IEEE, 2004.

[89] B. Srivastava and J. Koehler. Planning with workflows-an emerging paradigm for web service composition. In *ICAPS 2004 Workshop on Planning and Scheduling for Web and Grid Services*, 2004.

[90] P. Steggles and S. Gschwind. *The Ubisense smart space platform*. na, 2005.

[91] V. Stirbu. Towards a restful plug and play experience in the web of things. In *Semantic computing, 2008 IEEE international conference on*, pages 512–517. IEEE, 2008.

[92] T. Strang and C. Linnhoff-Popien. A context modeling survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

[93] W. Su, H. Eichi, W. Zeng, and M.-Y. Chow. A survey on the electrification of transportation in a smart grid environment. *Industrial Informatics, IEEE Transactions on*, 8(1):1–10, 2012.

[94] H. Sun, V. D. Florio, N. Gui, and C. Blondia. Promises and challenges of ambient assisted living systems. In *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*, pages 1201–1207. Ieee, 2009.

[95] T. S. Ustun, C. R. Ozansoy, and A. Zayegh. Implementing vehicle-to-grid (v2g) technology with iec 61850-7-420. *Smart Grid, IEEE Transactions on*, 4(2):1180–1187, 2013.

[96] M. Vallée, F. Ramparany, and L. Vercouter. Dynamic service composition in ambient intelligence environments: a multi-agent approach. In *Proceeding of the First European Young Researcher Workshop on Service-Oriented Computing*, pages 1–6. Citeseer, 2005.

[97] M. Vallée, F. Ramparany, and L. Vercouter. Flexible composition of smart device services. *PSC*, 5:165–171, 2005.

[98] D. Valtchev and I. Frankov. Service gateway architecture for a smart home. *Communications Magazine, IEEE*, 40 (4):126–132, 2002.

[99] J. I. Vázquez, D. López de Ipiña, and I. n. Sedano. Soam: A web-powered architecture for designing and deploying pervasive semantic devices. *International Journal of Web Information Systems*, 2(3/4):212–224, 2007.

[100] V. VideoLan. Media player. https://www.videolan.org/, 2007.

[101] S. Vinoski. Rest eye for the soa guy. *IEEE Internet Computing*, 11(1):82–84, 2007.

[102] S. Vinoski. Restful web services development checklist. *Internet Computing, IEEE*, 12(6):96–95, 2008.

[103] H. Viswanathan, B. Chen, and D. Pompili. Research challenges in computation, communication, and context awareness for ubiquitous healthcare. *Communications Magazine, IEEE*, 50(5):92–99, 2012.

[104] M. Vukovic and P. Robinson. *Adaptive, planning based, web service composition for context awareness*. na, 2004.

[105] M. Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[106] P. Wisner and D. N. Kalofonos. A framework for end-user programming of smart homes using mobile devices. In *Proceedings of the 4th IEEE consumer communications*

*and networking conference CCNC*, volume 7, pages 716–721, 2007.

[107] C.-L. Wu, C.-F. Liao, and L.-C. Fu. Service-oriented smart-home architecture based on osgi and mobile-agent technology. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(2): 193–205, 2007.

[108] E. Zeeb, A. Bobek, H. Bohn, and F. Golatowski. Lessons learned from implementing the devices profile for web services. In *Digital EcoSystems and Technologies Conference, 2007. DEST'07. Inaugural IEEE-IES*, pages 229–232. IEEE, 2007.