Studies in Computational Intelligence 460

Nik Bessis
Fatos Xhafa
Dora Varvarigou
Richard Hill
Maozhen Li (Eds.)

Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence

Springer

Studies in Computational Intelligence 460

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/7092

Nik Bessis, Fatos Xhafa, Dora Varvarigou,
Richard Hill, and Maozhen Li (Eds.)

Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence

Springer

Editors

Nik Bessis
School of Computer Science
and Mathematics
University of Derby
Derby
UK

Richard Hill
School of Computer Science
and Mathematics
University of Derby
Derby
UK

Fatos Xhafa
Departament De Llenguatges I
Sistemes Informàtics
Universitat Politècnica De Catalunya
Barcelona
Spain

Maozhen Li
Department of Electronic and
Computer Engineering
School of Engineering and Design
Brunel University
Uxbridge
UK

Dora Varvarigou
School of Electrical & Computer
Engineering
Division of Communication, Electronic
and Information Engineering
National Technical University of Athens
Athens
Greece

ISSN 1860-949X e-ISSN 1860-9503
ISBN 978-3-642-34951-5 e-ISBN 978-3-642-34952-2
DOI 10.1007/978-3-642-34952-2
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012951712

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

The past twenty years have witnessed unprecedented innovations in the development of miniaturized electromechanical devices and low-power wireless communication making practical the embedding of networked computational devices into a rapidly widening range of material entities. This trend has enabled the coupling of physical objects and digital information into cyber-physical systems and is widely expected to bring about ubiquitous computing. To be sure, this vision for future computing systems has matured from a curiosity of the academic laboratory into everyday reality for many in the short span of roughly two decades.

One of the core ingredients of this vision, the so-called Internet of Things, demands the provision of networked services to support interaction between conventional information systems and such augmented natural objects and manufactured artifacts, places, and a variety of embedded and mobile electronic devices. On the Internet of Things, physical entities produce and consume information, and participate in peer-to-peer and community activities as first class citizens. In the process, they generate and record detailed observations for every aspect of the physical world and the beings that inhabit it at a rate that far exceeds human-authored content such as the Web.

While microelectronics has played a critical role in providing the technological foundation for instrumentation and connectivity, the challenge for the next decade of the Internet of Things is how to record, manage and process this information so that it can be become useful. The focus of this book is exactly on the novel collective and computational intelligence technologies that will be required to achieve this goal. Specifically, it reports on methods and techniques that can be used to integrate, analyze, mine, annotate and visualize data captured on the Internet of Things. The contributions in this book explore alternative strategies towards achieving this objective such as data clustering, graph partitioning, collaborative decision making, self-adaptive and swarm intelligence and evolutionary agents. They also investigate how specific technologies such as social networks, the semantic web, knowledge representation and inference, cloud and peer-to-peer computing can address the unique challenges set in this context.

I believe that this book will be of great interest to all information and computer scientists and engineers, and all those in industry in academia working to transform the vision of a planetary-scale Internet of Things into the reality of tomorrow.

George Roussos
Birkbeck College, University of London, UK

Preface

Introduction

In recent years, resource providers have developed their own e-infrastructure platforms in a computationally isolated fashion, which are not necessarily inter-operable and inter-cooperative for effective data portability, service and resource sharing, discovery, scheduling and integration. Whilst inter-operable and inter-cooperative initiatives have been a particular area of interest for researchers, the trend towards 'inter-connected-ness' has fuelled a greater demand for work in this field. Specifically, rapid developments in networking and resource integration have resulted in the emergence, and in some instances,the maturation of various distributed and collaborative computational technologies including Web 2.0, SOA, P2P, Grids and Clouds. A number of relevant e-infrastructure implementations demonstrate the applicability of these technologies in a manner that enables improved intelligence in decision-making, through their agile and synergetic capacity, which in turn, seems a promising way forward for solving complex computational problems, and real-world grand challenges. These technologies are becoming even more popular as they provide improved utility, consumption, delivery and efficiency models for the Future Internet, the Internet of Things (IoT).

For example, Cloud computing has emerged as one of the most important technologies for delivering on-demand advanced services via the Internet. Like SOA, P2P and Grids, Clouds are also seen as a pre-cursor of the IoT. A variety of Cloud vendors including Amazon, Google and Salesforce develop their services by spreading them at different geographically locations, and by making them available for utilization across a worldwide set of Internet users. As the number of resource consumers increases, there is a need to establish behaviors relating to Quality of Service (QoS) for Cloud Computing architectures. The underlined inter-operable and inter-cooperative requirements (inter-Cloud initiative, also known as federated Clouds), highlight the current need for supporting a coordinated distribution of the workload between different Clouds, in particular (and e-infrastructures in general) and for the benefit of their Internet users. The inter-Cloud initiative goes beyond current Cloud capabilities by providing a flexible e-infrastructure towards an integrated solution supporting the requirement for improved inter-organizational functionality, which in turn will foster further organizational opportunities. The computational vision as a whole is to continue developing inter-functionality between e-infrastructures, that is to say, forming a pool of inter-operable and inter-cooperative sub-e-infrastructures that enables the dynamic collaboration of networked, inter-connected organizations.

Thus, one of the aims of this book is to discuss the progress made as well as prompt future directions on the utilization of inter-operable and inter-cooperative next generation computational technologies, which supports the IoT approach, that being an advanced functioning towards an integrated collective intelligence approach for the benefit of various organizational settings.

Apart from the inter-operable and inter-cooperative aspects, the book aims to deal with a notable opportunity namely, the current trend in which collectively shared and generated content emerges from Internet end-users. Specifically, the book aims to discuss advances about utilizing and exploiting data and opportunities generated from within inter-operable and inter-cooperative e-infrastructures towards an integrated, collective and computational intelligence approach. It is believed that the 'bringing together' functionality is somehow an advanced application of an integrated collective intelligence approach for the Future Internet.

Who Should Read the Book?

The content of the book offers state-of-the-art information and references for work undertaken in the challenging areas of IoT and various inter-cooperative computational technologies for the purpose of an integrated collective intelligence approach. Thus, the book should be of particular interest for:

Researchers and doctoral students working in the area of IoT sensors, distributed technologies, collective and computational intelligence, primarily as a reference publication. The book should be also a very useful reference for all researchers and doctoral students working in the broader fields of computational and collective intelligence, emerging and distributed technologies.

Academics and students engaging in research informed teaching and/or learning in the above fields. The view here is that the book can serve as a good reference offering a solid understanding of the subject areas.

Professionals including computing specialists, practitioners, managers and consultants who may be interested in identifying ways and thus, applying a number of well defined and/or applicable cutting edge techniques and processes within the domain areas.

Book Organization and Overview

The book contains 18 self-contained chapters that were very carefully selected based on peer review by at least two expert and independent reviewers. The book is organized into three sections according to the thematic topic of each chapter. The following three sections reflect the general themes that are of interest to the IoT community.

Section I: State-of-the-Art Critical Reviews

The section focuses on presenting state-of-the-art reviews on Internet of Things and applicable inter-cooperative and inter-operable methods and techniques to enable collective and computational intelligence. The section consists of 6 chapters. In particular:

Chapter 1 explores how the Internet of Things and the evolution of the Web as a highly interoperable application platform for connecting real-world things, has raised many research challenges, leading to the fast growing research area called the Web of Things.

Chapter 2 presents the challenges, state of the art, and future trends in context aware environments (infrastructure and services) for the Internet of Things. The chapter also describes relevant research work in the infrastructure, and up to date solutions and results regarding infrastructure and services.

Chapter 3 focuses on service interoperability models, which have been conceived for large grained hypermedia documents. Their extension to the Internet of Things context implies using a subset of those technologies. This chapter assesses the constraints and limitations of those technologies and prompts goals for their solution. A new interoperability technology is proposed.

Chapter 4 discusses the role of ad hoc networks in the Internet of Things. Within this context, the chapter offers a state-of-the-art review on ad hoc and wireless sensor networks, near field communications, radio frequency identification and routing protocols as a means to describe their applicability towards an Internet of Things realization.

Chapter 5 explains that the increasing pervasiveness in today's networks, leads to numerous efficiency and scalability challenges. The chapter details the state-of-the-art and binding concepts for efficient real-time sharing and mobility of multimedia and context.

Chapter 6 highlights the recent threats that malware pose towards RFID systems. The chapter describes a dual pronged, tag based SQLIA detection and prevention method optimized for RFID systems. The approach is composed from an SQL query matching and a tag data validation and sanitization techniques.

Section II: Advanced Models and Architectures

This section focuses on presenting theoretical and state-of-the-art models, architectures, e-infrastructures and algorithms that enable the inter-cooperative and inter-operable nature of the Internet of Things for the purpose of collective and computational intelligence, and consists of 6 chapters. In particular:

Chapter 7 explains that the Internet of Things will revolutionize the way businesses will interact, collaborate and transact with customers, suppliers, partners, employees and shareholders. The Chapter presents a conceptual model for performance analysis of software services availability vs. interoperability in order to optimize business processes at different levels.

Chapter 8 deals with critical infrastructures, such as the Smart Power Grid, which move beyond centralized management and control by system operators and administrators. The chapter illustrates Application-level Self-Organization Services for Internet-scale Control Systems model, that makes the Internet of Things inter-operation possible.

Chapter 9 is concerned with delay tolerant networks. The chapter presents and evaluates a utility-based routing protocol, which maximizes the expected number of selected relay nodes being likely to encounter a destination node under sequential encounters with other nodes. The performance of the proposed protocol is stable up to a few hundred mobile nodes.

Chapter 10 focuses on intelligent transportation systems technology. Within this context, the chapter describes a new architecture based on a virtual cloud computing environment for optimal scheduling of batch simulations, in a hybrid cloud environment.

Chapter 11 explains that large-scale infrastructures encounter challenges in relation to their scalability and interoperability. The chapter presents and proofs a large-scale multi-agent system architecture based on semantic P2P Network – Virtual Hierarchical Tree Grid Organizations (VIRGO) in mathematic terms.

Chapter 12 explains that with the advent of the Internet of Things, the P2P networks have found increased interest in the research community since the search protocols for these networks can be gainfully utilized in the resource discovery process for the IoT applications. The chapter presents a secure and efficient searching protocol for unstructured P2P networks that utilizes topology adaptation, by constructing an overlay of trusted peers and increases the search efficiency by intelligently exploiting the formation of semantic community structures among the trustworthy peers. Extensive simulation results are presented.

Section III: Cutting-edge and Future Applications

The section focuses on presenting cutting-edge Internet of Things related applications to enable collective and computational intelligence, as well as prompting future developments in the area. The section consists of 6 chapters. In particular:

Chapter 13 describes how the Internet of Things vision converges with the vision for Intelligent Environments (IEs), as ubiquitous computing deployments that are endowed with Ambient Intelligence. The chapter is concerned with the marriage of passive objects from the Internet of Things, and active-objects from IE as symbiotic if real-world deployment can ever be achieved.

Chapter 14 presents GENIUS as an on-going Internet of Things inspired project. GENIUS is concerned with the creation of a flexible architecture that is able to support a wide range of ‘intelligent’ applications, focused upon the recognition and interaction with the so-called Generalized World Entities (GWEs). The GWE paradigm intends to fill up the present fracture between the detection of entities at the sensor/physical level and their representation/management at the conceptual level. An “Ambient Assisted Living (AAL)” application for dealing with the “elderly at home problem” is presented.

Chapter 15 explains that as the amount of gathered data increases everyday, making the analysis of it a more complex task. The chapter focuses on text processing tools dealing with stemming algorithms and presents an Apache Mahout plug-in for a stemming algorithm making possible to execute the algorithm in a cloud computing environment. The performance evaluation of the approach reports that the execution time is significantly reduced.

Chapter 16 describes a context-aware recommender system that accommodates user’s needs with location-dependent multimedia information, available in a mobile environment related to an indoor scenario.

Chapter 17 describes the iCare approach as to provide better support to patients from the comfort of their home. iCare takes full advantage of Semantic Web technologies and Internet of Things and provides a new patient-centered approach, which significantly reduce the patient dependency from their doctors.

Chapter 18 is concerned with biometric authentication systems, which represent a valid alternative to conventional authentication systems, providing robust procedures for user authentication. On the other hand, Internet of Things involves a heterogeneous set of interacting devices to enable innovative global and local applications and services for users. The chapter focused on describing and contrasting fingerprint and iris based unimodal and multimodal authentication systems, as well as presenting a prototyped embedded multimodal biometric sensor.

Nik Bessis
University of Derby, UK

Fatos Xhafa
Universitat Politècnica de Catalunya, Spain

Dora Varvarigou
National Technical University of Athens, Greece

Richard Hill
University of Derby, UK

Maozhen Li
Brunel University, UK

Acknowledgements

It is with our great pleasure to comment on the hard work and support of many people who have been involved in the development of this book. It is always a major undertaking but most importantly, a great encouragement and definitely a reward and an honor when experiencing the enthusiasm and eagerness of so many people willing to participate and contribute towards the publication of a book. Without their support the book could not have been satisfactory completed.

First and foremost, we wish to thank all the authors who despite busy schedules, they have devoted so much of their time preparing and writing their chapters, and responding to numerous comments made from the reviewers and editors. Special gratitude goes to Dr George Roussos from Birkbeck College, University of London, UK for writing the foreword to this collection and also to the reviewers for their constructive commentaries and insightful suggestions.

Last but not least, we wish to thank Dr Thomas Ditzinger; Professor Janusz Kacprzyk and the whole of Springer's editorial team for their strong and continuous support throughout the development of this book.

Finally, we are deeply indebted to our families for their love, patience and support throughout this rewarding experience.

Nik Bessis
University of Derby, UK

Fatos Xhafa
Universitat Politècnica de Catalunya, Spain

Dora Varvarigou
National Technical University of Athens, Greece

Richard Hill
University of Derby, UK

Maozhen Li
Brunel University, UK

Contents

Section I: State-of-the-Art Critical Reviews

Section II: Advanced Models and Architectures

Section III: Advanced Applications and Future Trends

The Web of Things – Challenges and Enabling Technologies

Sujith Samuel Mathew, Yacine Atif, Quan Z. Sheng, and Zakaria Maamar

Abstract. The *Internet of Things* (IoT) is an active research area, focusing on connecting real-world things over TCP/IP. This trend has recently triggered the research community to adopt the interoperability of the Web (HTTP) as an application platform for integrating '*things*' on the Internet. Harnessing physical things into the virtual world using Web standards is also enriching the arena of conventional Web services to unleash data and functions of real-world things as service providers and consumers on the Internet. This evolution of the Web as a highly interoperable application platform for connecting real-world things has raised many research challenges and problems, leading to the fast growing research area called the *Web of Things* (WoT). Current research on WoT is a catalyst for the realization of IoT, opening up the possibilities of creating *ambient spaces* (AS), where people and things seamlessly communicate over the Web. In this chapter we discuss the state of the art in WoT research, focusing on the various challenges, and enabling technologies that are driving this research. We discuss architectural frameworks, models and technologies to build applications for future ambient spaces with the WoT. We present case studies that reflect the feasibility and applicability of the WoT technology. We also discuss future trends and research directions within this domain to throw light on existing problems and challenges.

1 The Web – An Application Platform for Real-World Things

Weiser envisioned *Ubiquitous Computing*, where computing becomes invisibly integrated into the world around us and accessed through intelligent interfaces. He observed, "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*" [1]. The recent technology advances in the area of communication and embedded systems are enabling the realization of this vision. The *Internet* is today the

Sujith Samuel Mathew · Quan Z. Sheng
School of Computer Science, University of Adelaide, Adelaide, Australia

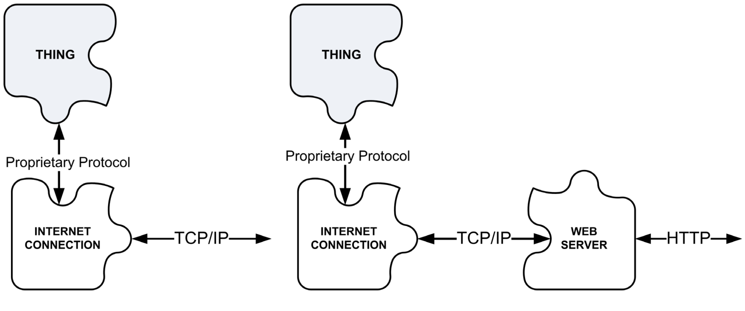
Yacine Atif
College of IT, UAE University, Al Ain, UAE

Zakaria Maamar
College of IT, Zayed University, Dubai, UAE
e-mail: {sujith,quanzheng.sheng}@adelaide.edu.au,
yacine.atif@uaeu.ac.ae, zakaria.maamar@zu.ac.ae

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 1–23.
DOI: 10.1007/978-3-642-34952-2_1

communication platform, which enables this fast-paced immersion of the physical world into the virtual world. Internet-enabled mobile phones, televisions, refrigerators, bathroom scales, and pacemakers are paving the way for everyday objects around us such as sidewalks, traffic lights, and other commodities to be identifiable, readable, recognizable, addressable, and even controllable via. the Internet (TCP/IP) [2]. These trends are motivating researchers to study, model, design, and implement novel applications, which promote interoperability of heterogeneous physical things on the Internet. The *Internet of Things* (IoT) is a fast growing area of research brimming with new ideas, discoveries, standards and challenges towards encompassing things into the virtual world of the Internet. However, networking alone does not enable the success or usability of IoT. Today, business and industry depend on applications that are built on Web architecture and on the interoperability of Web services. This paradigm is an additional focus of the IoT and research in this direction is termed as the *Web of Things* (WoT).

While the IoT provides the necessary networking and communication protocols to access real-world things, a parallel and recent research trend evaluates the Web as a platform for building applications that integrate real-world things on the Internet. The spread of the Internet provides the networking infrastructure for the use of real-world things, while research in the WoT provides the application and service layer for real-world things to interoperate over HTTP [3, 4]. The Web has been an important catalyst in the spread and popularity of the Internet. Its open platform has engaged application providers and users to generate and exchange information securely in various formats. Existing network infrastructures like Wi-Fi, Ethernet, and 3G leverage the use of Web technologies which facilitate the availability of tools, frameworks and information for developing Web based applications. These factors reduce the learning curve when using Web technology for extending the scope of Web applications to real-world things.



(a) Internet Enabled Thing (b) Web Enabled Thing

Fig. 1 Realizing a Web-enabled thing

A high-level representation to realize a Web-enabled thing is illustrated in Fig. 1. Augmenting a thing with Internet connection (i.e., IP address) ensures its accessibility over the Internet and results in an Internet-enabled thing, as shown in Fig.1 (a). When a thing is Internet-enabled and is, also, connected to a Web server, then it becomes Web-enabled, as shown in Fig.1 (b). The Web server may be either embedded or on a separated system. Advances in embedded technology have made it possible for realizing tiny Web servers which are widely available now and fueling the trend towards having Web servers embedded in everyday objects [5].

The wide proliferation of the Web has paved the path for Web services to be an indispensable technology, for interoperable applications. Extending service computing to Web-enabled things makes it possible for real-world things and their operations to be abstracted as Web services. In doing so, real-world things could offer their functionality via SOAP-based Web services (WS-*) or RESTful APIs [6]. These services directly expose operations of real-world things to enterprise applications or even to other things and possibly involving a dynamic composition process at runtime. This research direction of the WoT opens up many opportunities and challenges where real-world things are identified, searched, and composed on the Web. Therefore, the WoT facilitates the realization and adaptation of IoT, driving the possibilities of creating *ambient spaces* (AS), which are virtual spaces where people and real-world things seamlessly communicate over the Web.

In this chapter we present a detailed review of the state of the art, challenges, and enabling technologies of WoT that promote the IoT. We illustrate model architectures and discuss technologies that are enabling this research trend. We discuss future applications through case studies that realize the creation of AS with the WoT. The chapter concludes by providing insight into future research trends to throw light on existing problems and challenges.

2 Towards the Web of Things

The idea of integrating and networking real-world things to communicate with each other has been prevalent for many years. Various technologies and standards have been proposed and some are already in use today. Many of these technologies, both software and hardware, have paved the path for the adoption of WoT. The primary challenge is to address the *interoperability* of the tirade of things that surround us today. We present some of the early attempts that enabled communication of things on the Web and also look at technologies that are driving the research on the WoT.

Several industry alliances and standards have been defined like UPnP, DLNA, SLP, and Zeroconf to facilitate interoperability of real-world things. Each of these standards has individually been successful in enabling devices to communicate with each other. However, the availability of many standards creates an even more complicated challenge, which is *interoperability* between standards. With the diversity of things, their vendor specific properties, and the variety of services they provide, the use of standard-based interoperability approach is a never-ending process [7]. We briefly present two of these that have made considerable impact.

An early and noteworthy example is *Zeroconf* [8], which is a technique to provide reliable networking without any manual configuration and support services. Zeroconf is offered using several implementations. Probably the most known is

Apple's *Bonjour* which is used to discover shared printers for streaming media to Apple TV devices.

Universal Plug and Play (UPnP[1]) is a collection of networking protocols promoted by the UPnP forum and mainly used for devices to discover each other and establish connections over a common network. UPnP is based on protocols and standards like HTTP, HTTPU (i.e., HTTP over UDP), UDP, TCP, and IP. UPnP supports various services like discovery, description, control, event notification, and presentation. However, this cannot be widely used because in some embedded devices that are resource constrained it is impossible to host a large number of these protocols [9, 10]. Conventional middleware technologies such as CORBA and Java-RMI have been used to realize device interoperability by standardizing common protocols. The *Jini* project created a middleware where each device is loosely coupled as a Java object. Allard et al. introduced a framework to combine Jini with UPnP [11].

2.1 Technologies Driving the Web of Things

There are an increasing number of Internet-enabled things flooding the markets. Alliances such as the IP for Smart Objects (IPSO[2]) and the European Future Internet Initiative (EFII[3]) have accelerated this trend to connect a variety of physical things to the Internet, with the intention of propagating the wide use of Internet as the common medium for communication. Also, with the ever-decreasing size of embedded systems and related software footprint, it has become possible to directly integrate Web servers into many appliances [5]. This trend to use the Web as a platform to create and integrate applications that integrate real-world things into the Web has attracted attention in academia and businesses [4, 6, 31]. Service Oriented Architecture (SOA) has proven to be a promising for integrating devices into a business IT network [32]. This has in turn led to the propagation of using *Web services* for the interoperability of real-world things.

2.1.1 Web Services

The use of Web services is paramount in establishing interoperable distributed systems on a network. This indispensable technology is extended to WoT to allow data or functionalities of real-world things to be abstracted and exposed as services on the Web. There are two major classes that regulate the proliferation of Web services, the RESTful Web services and the WS-* protocols stack [12, 13].

RESTful Web Services

RESTful Web services are based on Representational State Transfer (REST) [13], an architectural style that considers every participating component to be an addressable *resource*. The state of each resource is essentially captured in a

[1] `http://www.upnp.org/`

[2] `http://www.ipso-alliance.org/`

[3] `http://www.future-internet.eu/`

document referred to as a *representation*. The Web was built upon this architecture where each resource is a Web page identified by a URI (Uniform Resource Identifier). REST is lightweight, simple, loosely coupled, flexible and seamlessly integrates into the Web using HTTP as an application protocol. RESTful Web services are based on the following four concepts [14].

- *Resource identification*: Resources are identified through URIs which provide a unique and global presence for service discovery.
- *Uniform interface*: Resources are accessed using the standard HTTP verbs, GET, POST, PUT, and DELETE.
- *Stateful interactions through hyperlinks*: Interactions with resources are stateless and self-contained. To circumvent this, state can be embedded into response messages to point to valid future states of the interaction.
- *Self-descriptive messages*: Resources are decoupled from their representation so that their content can be accessed in a variety of formats like Extensible Markup Language (XML), JavaScript Object Notation (JSON), plain text or YAML Ain't Markup Language (YAML).

Though REST has been used to build the traditional Web as we know it, these same attributes made REST an ideal platform to expose services of real-world thing on the Web [27, 28]. IETF has constituted a working group called *Constrained RESTful Environments* (CoRE[4]), to develop a framework for resource oriented applications intended to run on constrained IP networks like WSN (Wireless Sensor Networks). This includes sensor based applications like temperature sensors, light switches, and power meters, applications to manage actuators like light switches, heating controllers, and door locks, and applications that manage devices. As part of the framework for building these applications, CoRE plans to define Constrained Application Protocol (CoAP) for the manipulation of resources on devices. CoAP is to be designed for interaction between devices on traditional IP networks, on constrained IP networks, and between constrained networks on the Internet. CoAP is a work in progress but some implementations are already emerging such as the Firefox extension Copper[5], Tiny OS[6], and libcoap[7]. The CoRE group has proposed the following features for the CoAP:

- RESTful design minimizing the complexity of mapping with HTTP,
- Low header overhead and parsing complexity,
- URI and content-type support,
- Discovery of resources provided by CoAP services,
- Subscription for resources and resulting push notifications, and
- Caching mechanism.

A REST-based approach was, also, used to define the TinyREST architecture [15], which introduces a new HTTP method, SUBSCRIBE, that enables clients to

[4] http://datatracker.ietf.org/wg/core/charter/

[5] https://addons.mozilla.org/en-US/firefox/addon/copper-270430/

[6] http://docs.tinyos.net/tinywiki/index.php/CoAP

[7] http://libcoap.sourceforge.net/

register their interests to device level services. The pREST or pico-REST [16] proposed by Drytkiewicz et al. is an access protocol, which brings the REST concept to the device level. The emphasis is on the abstraction of data and services as resources in sensor networks.

Though the work on REST-based services continues to encompass many WoT applications, in enterprise applications like banking or stock markets, where Quality of Service (QoS) levels are stricter, a more tightly coupled service paradigm like WS-* would be more ideal [14].

WS-* Based Services

The key enabling technologies of WS-* are Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), and Universal Description Discovery and Integration (UDDI). The services use HTTP as a transport protocol over which SOAP messages are sent. SOAP [17] is an XML-based protocol defining a message based architecture and format. SOAP messages are contained in an XML element called *envelop*, which contains two sub-elements called *header* and *body*. The *header* features application specific information and also QoS parameters while the *body* contains the content of the message intended for the endpoint. WSDL [18] is also an XML-based language describing the syntactic communication for message exchange between endpoints. SOAP messages and their structures are described by WSDL. Service endpoints are addressed either on the transport level, i.e., with Universal Resource Identifiers (URIs) for SOAP/HTTP-bound endpoints, or on the messaging level via WS-Addressing [19]. UDDI [20] is also an XML-based registry framework for discovering Web services. While this technology has been mature and stable for several years now, it has failed to reach widespread acceptance in the industry.

WS-* is stateless but WS-Resource Framework [21] describes the interaction of Web services where the state needs to be maintained. The WS-* technology stack also covers many QoS features required to ensure the interoperability of distributed systems [22]. A large set of WS-* specifications has been suggested because, this approach enables the composition of Web services and is, also, flexible to be used by real-world things. Research on integrating WS-* into real-world things has been active for many years. A Service-Oriented Device Architecture (SODA) [23] is proposed to integrate real-world things into today's businesses. The architecture exposes device functionalities as abstract business services. Pintus et al. [24] describe an SOA framework where real-world things are described using WSDL. The communications between these things are modeled as service orchestrations using BPEL (Business Process Execution Language). Projects such as SIRENA (Service Infrastructure for Real-Time Embedded Networked Applications) [25] and SOCRADES (Service Oriented Cross-layer Infrastructure for Distributed Smart Embedded devices) [26] adapt an approach for networks with smart things. We discuss the SOCRADES Integration Architecture, later in the chapter.

Merging WS-* and RESTful Web Services

Since the goal of both paradigms is the same, there have been attempts of merging the two service architectures for WoT to make up for the disadvantages of each

[27, 28]. The first direction is to have the RESTful architecture merged into the existing WS-* service architecture [27], which requires the implementation of alternate data model. Having two data models for the same service results in a very complex architecture and is not desirable. On the contrary the other direction is to have a software module that translates RESTful requests into WS-* requests [28]. This provides a structured approach ensuring a REST centric architecture while maintaining the robustness of WS-* approach, but this affects the performance of RESTful APIs.

2.1.2 Embedded Web Servers

Embedded Web servers are an indispensable component for the long term adaptation and spread of WoT. These tiny Web servers enable communication between real-world things over the simple and widely used HTTP protocol. Researchers have successfully embedded tiny Web servers into resource-constrained things like sensors and smart cards, making Web-enabled things a reality [29, 30]. Filibeli et al. [33] describes an embedded Web server based home network system where Ethernut[8] based Web servers are embedded into home appliances and are controlled via the Web. Priyantha et al. [34] proposes the interoperability of sensor nodes by implementing Web servers on the nodes. Today, open source tiny Web server modules like FlyPort[9] from OpenPicus are readily available off the shelf. These modules are 35X48 mm in dimension, comes with an integrated 802.11 b/g/n Wi-Fi interface module, and a 16-bit processor. The internal flash of 256KB has been demonstrated to be sufficient for different Web applications that access and control real-world things like actuators and sensors. With the reducing physical size of hardware components and its software footprint many of the new appliances in the market, like Smart TVs and refrigerators are expected to come with embedded servers. This trend indicates that in the near future, embedded Web servers will be common feature in many of the physical things that surround us, enabling intelligent WoT applications.

2.1.3 RFID Tagged Things

Research and development on the IoT gained its recent momentum from the wide spread deployment and use of RFID (Radio Frequency IDentification) technology. As illustrated in Fig.2, information of a *tagged thing* like a can of beans, become available on the Internet when it is tagged with an RFID tag. An *RFID reader* reads the information on a tag and an *Application server* uses the reader's access protocol to access the information of the tags and expose the information on the Internet.

It is important to clarify the term *things* while considering WoT or IoT applications and systems. While the term *devices* or *appliances* are often used to refer to things under consideration, some tend to believe that almost every "thing" can be included. The rationale for the latter is that information of any *tagged thing* is

[8] `http://www.ethernut.de/`

[9] `http://www.openpicus.com/`

accessible on the Internet. The former argues that only *connected devices* on the Internet fall into the category of *things*, for example, in Fig.2 the Application server is a *connected device*. Today, RFID readers are Internet enabled, and directly access the Internet without the use of an intermediate server and has been elevated into the category of *connected devices* or *connected things*.

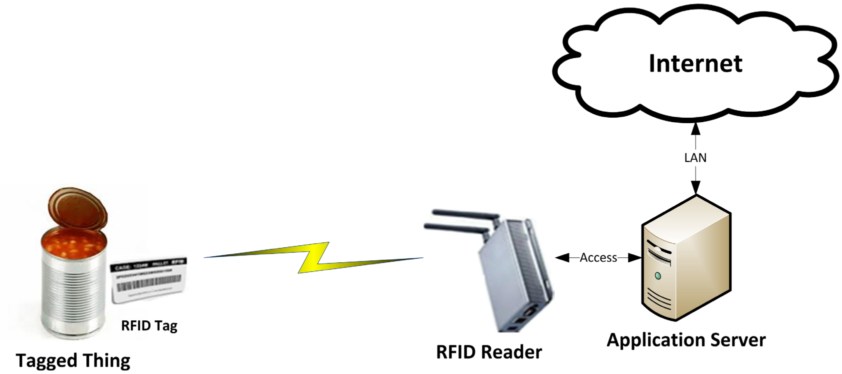


Fig. 2 Internet enabling things with RFID

With the advances in technology the capabilities of real-world things have been increasing and this enhancement has introduced the term *Smart* to be appended to these things. Thus, we have Smart phones, Smart TVs, and Smart Homes. The proliferation of *Smart* things with Web capabilities is driving the possibilities of realizing the WoT. Kortuem et al. [35] address issues on modeling and representing *Smart* objects and also striking a balance between the objects and the infrastructure. They classify Smart thing as: Activity-aware objects, policy-aware objects, and process-aware objects. These types represent combinations of three dimensions with the aim to highlight the interdependence between design decisions and explore how these objects can cooperate to form IoT. Mathew et al. [36] proposed an ontological approach to classify and manage real-world things within ambient spaces where the requirement of additional capabilities uses an ontology, which recognizes four fundamental dimensions of candidate elements. These dimensions of *Identity (ID), Processing, Communication, and Storage* are referred to as the IPCS set. The taxonomy refers to things as *Core* when it has only an ID and also facilitates the process by which things are made *Smart* by augmenting all four (IPCS) capabilities. These smart things have the potential to participate as Web resources on the WoT to provide information and enable interoperability between applications.

With more things tagged and accessible on the Internet, a major challenge is the management of large scale deployment of tagged things. These things are to be tracked and their state (e.g. location) is to be continuously updated. Discovering information like, current state, past state, as well as estimating the future states are important for the success of businesses that use them. Traceability is essential to a wide range of important business applications such as manufacturing control, logistics of distribution, product recalls, and anti-counterfeiting. Research in this direction has inspired the PeerTrack platform [61] which has several traceability applications built for mobile asset management and supply chain management for tagged things. A system for mobile asset management is deployed at the International Linen Service PTY LTD. (ILS), a company that provides a suite of linen services for over 200 customers in South Australia. In this system, trolleys are reusable containers for linens and are attached with RFID tags. They are transported among locations and are detected by RFID readers on arrive at a delivery location. The system developed on the PeerTrack platform offers an automated tracking and tracing service with the capability to monitor and control logistical operations in real-time over 300 different locations. Also, a visual monitoring tool is deployed at each customer's site, together with the P2P services. The system traces and tracks the mobility of the trolleys, and ensures real-time inventory monitoring.

Web technology has matured over the years to successfully manage the interoperability of systems distributed on a worldwide scale. The adaptation of tagged things and their respective information to the Web poses many challenges and research questions. The advances in WoT technology intends to provide a scalable, simple and foolproof platform for the management of the ever-increasing presence of tagged things.

2.1.4 6LoWPAN

Interoperability of things would only be possible on a scalable, accessible, and end-to-end communication infrastructure. To enable embedded Web server on devices, they must first be connected to the Internet (Fig.1.). Things with inherent information, that has to participate in a Web based application, are of various types and capabilities. The IP protocol stack should be adapted for devices with limited capabilities. *6LoWPAN* (IPv6 over Low power Wireless Personal Area Networks) was launched by IETF which defines mechanisms that allow IPv6 packets to be sent to and received between resource constrained devices usually by adopting low-power radio communication protocols like IEEE 802.15.4.

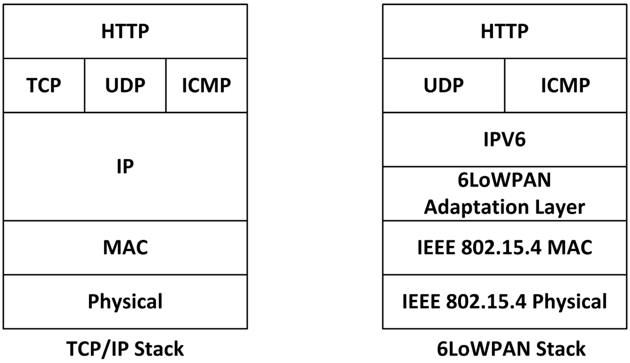


Fig. 3 Comparison between TCP/IP and 6LoWPAN protocol stacks

A layer-wise comparison between the TCP/IP and the 6LoWPAN stacks is shown in Fig.3. Zeng et al. in their survey of WoT [37] note that the necessity of an adaptation layer in the 6LoWPAN stack is mainly to fit one IP packet within one 802.15.4 MAC (Layer 2) frame. The adaptation layer also manages header compression, packet fragmentation, reassembling and edge routing. 6LoWPAN is ideally implemented using UDP as transport protocol which ensures efficiency when dealing with Internet-enabled things. Web applications will use HTTP over UDP and hence will be robust due to the unreliability of UDP. 6LoWPAN also makes things on the Internet addressable at the IP layer.

3 Web of Things Architecture

An architecture for the WoT should take into consideration the methods for integrating the plethora of things on the Web. It should formulate an abstraction framework that would make the heterogeneous capability of real-world things accessible and interoperable. We consider the ongoing efforts to realize an architecture for the WoT with these focuses.

3.1 Integrating Things on the Web

As illustrated in Fig. 1 (b), the requirement of Web-enabling a thing is to expose its operations as HTTP URLs. We explain two ways this can be achieved, (1) Direct Integration: augmenting the Web server to the thing, or (2) Indirect Integration: expose operations of a thing through a proxy Web server [6].

3.1.1 Direct Integration: Augmenting a Web Server

A thing is Internet-enabled and Web-enabled by augmenting it with the capability to communicate over TCP/IP and HTTP respectively. An embedded Web server exposes the thing's operations via URLs. When using RESTful APIs, a thing's operations are exposed through standard Web operations like GET or POST. Ostermaier et al. [5] present a prototype using Wi-Fi modules for connecting things directly to the Web. They enabled association of sensors, actuators, and things with each other through the Web. Guinard and Trifa [6] present the direct integration of *Sun SPOT*[10] with a Web server. They implement the embedded server directly on the Sun SPOTs nodes. The server natively supports the four main operations of the HTTP protocol i.e., GET, POST, PUT, and DELETE, making the Sun SPOT Web-enabled. Akribopoulos et al. [38] introduce a Web service-oriented architecture for integrating small programmable objects in the WoT. Using Sun SPOT applications and sensor data exposed through Web services, they present use cases for building automation and remote monitoring.

[10] `http://www.sunspotworld.com/`

3.1.2 Indirect Integration: Using a Proxy Web Server

The process of indirect integration involves a gateway or proxy server between the thing and the Web. A thing could have potential information that needs to be made available on the Web but does not have the necessary capabilities to communicate over TCP/IP or HTTP. The minimum requirement for such things to be an eligible candidate for the WoT is that they must be uniquely identifiable within a particular context. Mathew et al. [36] refer to such a thing as a *Core* thing and examples of such things would be pallets, medicine bottles, or shoes, which can be identified uniquely on the Web using an identification system like RFID or Barcode (Fig.2.). The information of *Core* things are made available to the Web through the proxy or gateway server, abstracting the proprietary protocol with uniform Web-based APIs.

Trifa et al. [39] implement a gateway for Web based interaction and management of sensor networks through RESTful Web services. Guinard et al. [40] build an EPC Network prototype by using virtualization, cloud computing, and Web technologies. In their prototype, the RFID reader behaves like a gateway which locates between the cloud server and RFID tags. Welbourne et al. [41] develop a suite of Web-based tools and applications that facilitate the understanding, management, and control of personal RFID data and privacy settings. They deployed these applications in an RFID ecosystem and conducted a study to measure the trends in adoption and utilization of the tools and applications as well as users' reactions.

3.2 Frameworks for the Web of Things

The use of Web services (WS-* and RESTful) is fundamental to any WoT architecture and various experiments have been tested and deployed. We look at two such architectures [26] and [28] to understand the layers used to expose things on the Web.

3.2.1 SOCRADES Integration Architecture (SIA)

The SOCRADES Integration Architecture (SIA) [26, 42] describes the use of both WS-* and RESTful services to access devices from enterprise applications.

The architecture includes a *local/on-premise subsystem* and a *central or remote or cloud subsystem.* A high-level illustration of the main layers of SIA with the modules of the local subsystem which interfaces with the devices is shown in Fig.4. SIA enables the integration of services of real-world things running on embedded devices with enterprise services. WS-* Web service standards constitute the main communication method used by the components of enterprise-level applications. This enables business applications to access real-time data from a range of devices through abstract interface based on Web services. SIA, also, supports RESTful services to be able to communicate with emerging Web 2.0 services. This enables any networked device that is connected to the SIA to directly participate in business processes.

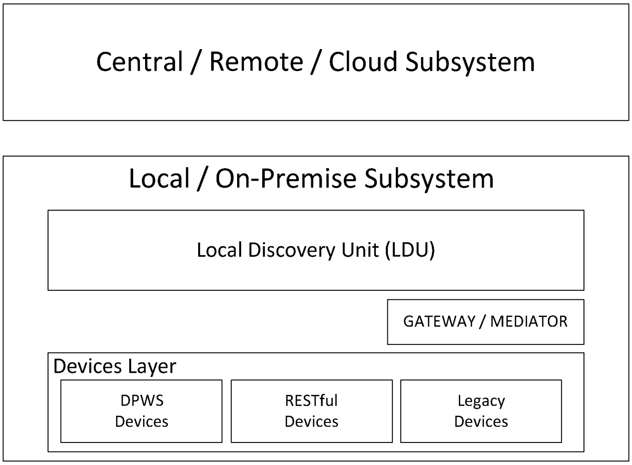

Central / Remote / Cloud Subsystem
Local / On-Premise Subsystem
Local Discovery Unit (LDU)
GATEWAY / MEDIATOR
Devices Layer
DPWS Devices
RESTful Devices
Legacy Devices

Fig. 4 Main layers of SOCRADES Integration Architecture (SIA)

The *Local/On Premise subsystem* features a *Local Discovery Unit* (LDU) connected to devices seen on a LAN, and the *Central subsystem* (anywhere on the network) hosts enterprise-level applications. The LDU module scans the local network for devices and reports their connecting and disconnecting to the central system. It acts as an intermediary that provides uniform access to different classes of devices through a set of platform-dependent plugins. In the local subsystem at the *Devices Layer* there are several embedded devices that run various services. The legacy devices would require a *Gateway* to expose their proprietary functionalities.

SIA is able to interact with devices using several communication protocols, such as DPWS (Device Profile for Web Services), OPC-UA, and REST. The details of each layer and their functionalities are provided in [26].

3.2.2 The Web of Things Application Architecture

The WoT application architecture presented by Guinard [28] enables the integration of things with services on the Web and facilitates the creation of Web based applications that operate on real-world things. The primary goals of the architecture is to enable rapid prototyping of applications that integrate real-world things for developers, to offer direct Web based access to things to Web users, and to offer lightweight access to things data which would enable the data to be consumed by other things or software.

In the proposed layered architecture, each layer does not hide the lower layers but rather provides a hierarchy of separate abstractions to connect things as first class citizens of the Web [28]. Based on requirements, an application can be built on top of each layer or a combination of layers. The various layers and their roles are illustrated in Fig.5, we present a summary of these layers here; a detailed representation is given in [28].

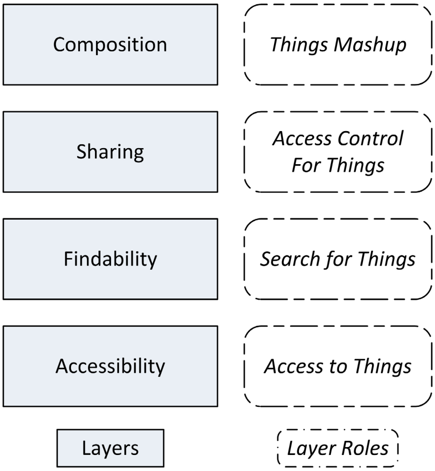


Fig. 5 Application Architecture for the Web of Things

The *Accessibility* layer focuses on providing a standard access to all kinds of connected things. This layer exposes things as *RESTful Things* using resource oriented architecture. REST [13] is used as a universal technique to interact with real-world things and four steps are suggested to Web-enable things:

1. Design the resources by identifying the functionality or services of a thing and organizing their interactions,
2. Design the representations and decide on how these will be provided,
3. Design the interfaces which indicate the actions of each service, and
4. Decide on direct or indirect integration into the Web.

Guinard [28] indicates that a client-initiated interaction model, where a Web client continuously polls a Web-enabled thing, is costly when dealing with resource constrained things. Instead a server-initiated model is proposed. Here, a Web client first registers with a Web server and is notified when an event occurs. This creates a real-time approach when a Web client interacts with a Web-enabled thing hosting a Web server. Moreover, the use of *tPusher,* which adds the use of *WebSockets*[11] to the WoT architecture, is suggested. tPusher moves the protocol to a gateway rather than a real-world thing [40]. This layer ensures that real-world things are exposed as first-class citizens on the Web. The *Findability* layer focuses on making services of real-world things searchable and available for service consumption or composition. Integration of things to the existing search engines with the use of semantic annotations is studied and the shortcomings of this approach were presented. The shortcomings are because of the inherent nature of real-world things, i.e., real-world things do not have properties that can be indexed like documents on the Web and things are tightly coupled with their contextual information like the owner of a thing, or its location. A lookup and registration

[11] http://dev.w3.org/html5/websockets/

infrastructure adapted to WoT is proposed. The *Sharing* layer focuses on how Web-enabled things are shared ensuring privacy and security. The requirements stated for a sharing platform for WoT are security, ease of use, reflection of trust models, interoperability and integrated advertisement. The architecture proposes the use of existing social networking platforms like Facebook, Twitter, and LinkedIn, as sharing hubs for things. Since these platforms already manage access control and trust measures, these can be leveraged to manage the access and privacy of things privately. The *Composition* layer focuses on enabling end-users to create simple composite applications with Web-enabled things. Web 2.0 mashup techniques are reused to create Physical mashup editors considering the requirements of the real-world things. The requirements are, support for event-based mashup, support for dynamic building blocks, support for non-desktop platform and support for application specific editors [43].

4 Building 'Web of Things Applications

Existing tools, techniques, languages and models for building Web applications can be applied to the WoT. However, for applications on resource constrained things where ad-hoc and event driven interactions are necessary, the challenges need to be understood and some techniques need to be adopted. We present some of these key techniques here.

4.1 The AJAX Technique

The AJAX (Asynchronous JavaScript and XML) technique [44] for Web application development creates highly interactive and efficient applications with efficient workload distribution between the client and the server modules. The behavior of an AJAX Web application can be separated into two phases: loading phase and running phase [29]. In the loading phase the client collects files that contain style, content, and application code, while in the running phase, the client executes the application code from the first phase, and interacts with the server by sending asynchronous requests, allowing the Web page to update itself dynamically. Experimental results [29] show that during the first phase, numerous large-sized content are served and during the second phase smaller size content is sent to the client. The AJAX technique uses the browser to handle the workload and relieves the Web server. This is an interesting technique for the WoT applications, where the embedded Web servers have lesser resources when compared to the clients.

4.2 Mashups - Composing Things on the Web

Mashups have eased the process of developing WoT applications. This new technique introduces a paradigm shift in the creation of Web applications by using Web 2.0 techniques and composing various Web services to achieve the application goals. Unlike portal-based solutions, the content aggregation for mashup applications can occur either on the client or on the server. This is beneficial for the WoT applications, where all the real-world things are abstracted as Web resources, which are addressable, searchable, and accessible on the Web. Guinard et al. [6], present two representative mashup styles, physical-virtual mashups and physical-physical mashups. For the former, the *Energy Visualizer* application was developed which has a Web interface to monitor power consumption and to control different home appliances. For the latter, to present the composition of services offered by things, the *Ambient Meter* was implemented on Sun SPOTs, which polls predefined URLs to get the energy consumption of devices in a room using HTTP based requests and responses. We discuss the Ambient Meter case study later in the chapter.

4.3 Event Driven Approach

The limitation of resources on embedded Web servers is a challenge that needs to be considered when creating applications for the WoT. Hence, a feature that is important to WoT application is the need to push data from server to client, based on events instead of the client continuously requesting to pull data from server. Event driven approaches are efficient to implement stateless behaviors and hence software designed for embedded systems ideally use event-driven approaches to manage resource constrained hardware.

Comet [45] allows a Web application to push data from the Web server to the client. Comet[12] is an umbrella term for techniques which are used for this interaction and implementation methods are of two types: *Streaming* and *Long Polling*. In *Long Polling*, the client polls the server for an event by sending requests and stops only when the server is ready to send some data. After it gets a response the client will restart a Long Polling request for the next event. In *Streaming*, the client opens a persistent connection from the browser to the server for all events. Neither side closes the connection and in the absence of events servers periodically send dummy data to keep the connection active. While these workaround methods are in practice they pose two drawbacks, they generate unnecessary traffic and they are extremely resource demanding on the Web servers. While Comet is better in data consistency and managing traffic workload, it has scalability issues.

More recently, Web Sockets were proposed which use full duplex communication with a single TCP/IP socket connection. It is accessible from any compliant browser using simple JavaScript API. The increasing support for HTML5 in Web and Mobile Web browsers makes it a very good candidate for pushing data for the WoT. Since WebSockets protocol has an initial handshake followed by message framing, layered over TCP, they can be implemented in a straightforward manner on any platform supporting TCP/IP [40].

5 Case Studies of Ambient Spaces

As stated earlier, the greater vision for the WoT is the creation of ambient spaces where people and things seamlessly communicate to achieve sustainable and

[12] `http://en.wikipedia.org/wiki/Comet_(programming)/`

efficient environments. We present two case studies that use WoT technology to realize AS.

5.1 Ambient Classrooms

Campus communities are notorious for their huge energy consumption despite awareness campaigns to reduce this consumption [46]. With the intention of creating sustainable campus environments we propose a framework which abstracts physical units like classrooms into AS or *ambient classrooms* on the Web. The framework makes these classrooms aware of the energy footprint of the users that reserved the room and also manage the usage of things like lights and projectors. The proposed framework was evaluated and it is noticed to save considerable amount of energy when compared to earlier usage scenarios.

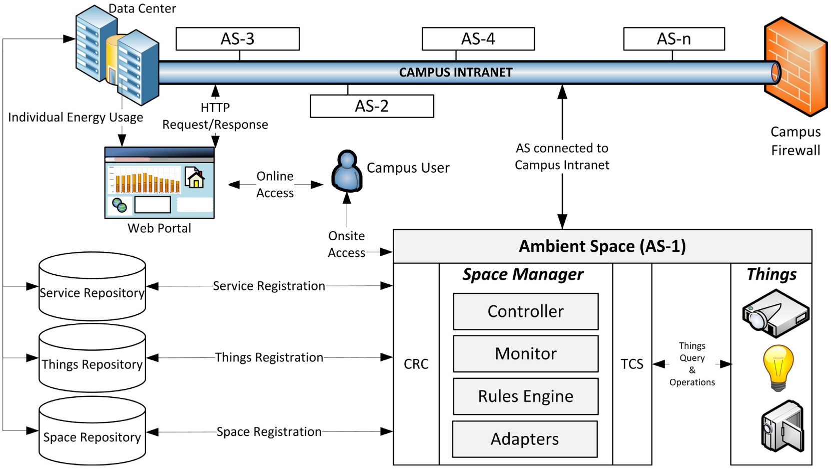


Fig. 6 Classrooms abstracted as ambient spaces on a campus Intranet

The framework illustrated in Fig. 6, adopts a service-oriented approach to connect classrooms as *Ambient Space* (AS) units on a campus. All classrooms on campus have similar things like, projectors, lights, and cameras. Each AS unit has a *Space Manager* comprised of hardware and software modules that enable things to communicate on the Web. The *Controller* provides Web services that grant operation of things within the AS. Things in the AS that are not Web enabled are augmented with Web capabilities using *Adapters* to provide the desired functionality. The *Rules Engine* constrains thing's usage based on operational intelligence which are specific to different types of physical things in AS. The *Monitor* provides interfaces with sensors that capture the status of AS for various measurements. Each AS has a designated set of repositories which is part of the campus *Data Center*. The *Space Repository* holds details like, location, purpose, and seating capacity, of all AS on campus. The *Things Repository* forms a knowledge base of all things associated with each AS and the *Service Repository* provides a directory of services exposed by each AS. The functionalities for the Space Manager are provided through two subsystems, the *Things Control and Sensing* (TCS) subsystem and the *Classroom Reservation and Control* (CRC) subsystem. We consider *lights* in the classrooms as candidates of Web-enabled things for this discussion.

The TCS subsystem comprises of Web-enabled things that are either augmented with embedded Web servers or indirectly connected to a proxy Web server. The TCS provides direct Web based access (RESTful APIs) to actuators like light bulbs in the room and is designed as the last unit of information flow as it may have the least computing resources. Lights have two basic operations, *on* and *off*. The challenge is to Web-enable inanimate things like light bulbs in order to push and pull information. Their capabilities are to be enhanced so that they can be controlled, sensed and their operations are accessible over the Intranet. We recently proposed an ontological approach to classify physical things within an AS environment and manage them from the Web [36]. The taxonomy of things facilitates the process by which things are made *Smart* by augmenting their IPCS capabilities, as mentioned earlier. The capabilities of *Smart Lights* are satisfied by:

- Identifying lights uniquely over the Intranet (IP address) and the Web (URL).
- Augmenting processing capability so that Web requests are processed and relevant actions are converted into electrical operations.
- Enabling communication capabilities to lights so that user actions and sensing results are communicated to and from the Smart lights.
- Providing storage capabilities for the lights to cache Web resources and status information.

This is realized by augmenting the classroom lights with tiny Web servers[13] and light sensors. TCS exposes two *Smart Light* services using RESTful expressions, *Light-ControlWS* to switch lights on or off and *Light-SenseWS* to detect the status of lights.

The CRC subsystem manages scheduling functions for the Ambient Classrooms and also maintains the rules for controlling things. Interactions with the Ambient Classrooms are provided through the CRC implemented on a digital classroom signage. The signage displays details of ongoing sessions and uniquely identifies a classroom. It is connected to the Intranet and accessed both online and onsite using a touch screen interface. The CRC hosts a *Calendar Service* that allows the scheduling of classrooms for courses. A genuine user with a valid username and password or a Web application like a time tabling application can reserve classrooms through CRC based on available time slots. CRC also hosts the *Ambience Service* to manage the state of things inside the classrooms depending on preset rules.

Rules are set to turn lights on or off based on room usage and also to monitor lights to determine if any of the lights need to be replaced. Classroom users are made aware of their energy usage based on the duration of time they use the

[13] `http://www.openpicus.com/`

classrooms. Each user monitors their individual *Energy Account* through an online Web portal, and incentives for optimal usage are planned.

5.2 Ambient Meter

The Ambient Meter is a prototype implementation that demonstrates the composition of real-world services of things using RESTful APIs. It specifies how things can interoperate to create new systems over HTTP. Guinard and Trifa [6] deployed this system on a mobile device that monitors the energy usage of all appliances in its vicinity. The Ambient Meter is sensitive to its location and it automatically adapts to the place it monitors, without human intervention. Based on the amount of energy consumption in the room, the Ambient Meter changes color to display the variations in energy footprint, from very green (energy consumed is low) to very red (energy consumed is high). Each room has various *Ploggs*[14] that are connected to appliances, like kettle, lamp, and PC. Their energy consumption is measured and processed by a *Smart Gateway*, which communicates with the Ploggs. A Smart gateway is co-located with the Ploggs in a room, which discovers the Ploggs and records the total energy consumed. The Smart Gateway is deployed alongside with an LLDU (Local Lookup and Discovery Unit) for resolving current location [6, 28]. When the mobile Ambient Meter (RESTful Sun SPOT) is located in a room it connects to the gateway in that room and becomes part of the local network. As the meter polls the gateway using a standard URI, a JSON representation of the energy consumption of all the Ploggs in the room is received. When the meter moves to another location it connects to the gateway in that room but uses the same URI to get the energy consumption recorded by the gateway in the room. Using RESTful approach the integration of the devices were reduced to building a Web mashup, where all the services are invoked by means of simple HTTP requests.

6 Research Directions

The WoT is transforming IoT technology just as the Web has transformed computing over the last decades. WoT enabling technologies are maturing and creating new platforms for designing and realizing innovative applications. We discuss some of the challenges and ongoing research towards the realization of the WoT vision.

Research has been focusing on adapting traditional client-server Web approaches to expose the functionalities of physical things [28]. This works fine when clients initiate interactions with servers to pull data and services from embedded applications that are expected to control things on the WoT. However to monitor things in AS, the required applications are often event-based and hence, Web-enabled things should also be able to push data to clients. Hence, standards such as HTML5[15] are moving towards asynchronous bi-directional communication to enable such interactions, where the server initiates events [28].

[14] `http://www.plogginternational.com/`

[15] `http://dev.w3.org/html5/eventsource/`

The spread of WoT is expected to flood the Web with real-time and real-world information and services that need to be discovered [47]. The use of sensors and other probing devices result in the dissemination of a large amount of real-time information. In contrast to the documents on the traditional Web, real-world things on the Web are expected to rapidly generate dynamic content because of their changing state. WoT applications inherently depend on the context of things and hence, search engines for WoT should focus on efficiently probing real-time data and discovering dynamic services. The efficiency and performance issues of search engines and related algorithms form some of the current areas of WoT research. Recently, Mayer et al. [48] suggested a discovery service for Web-enabled things, where users employ RESTful interfaces for discovering Web resources. A key challenge is to have an efficient search engine, which is able to reach and retrieve properties of real-world things on the Web. We presented some of these properties in light of the dynamic nature of real-world things [49].

Research is also directing focus towards optimizing communication protocols for resource constrained devices. Efforts to adapt application layer protocols like CoAP (Constrained Application Protocol) [50] based on REST and HTTP are active research areas, to enable flexibility and ease of use for a Web user and not just technical people. Refining protocols like 6LoWPAN [50, 51] that adapt the IPv6 protocols for low power radio networks is, also, highly relevant.

A key research challenge is the study of smart things composition to create context-aware AS [52]. With the plethora of real-world things that can be participants on the Web, this challenge is sophisticated and many innovative solutions are yet to be proposed. Research is focusing on the study of consumer reaction and trends in applications for eHealth, smart homes [53, 54], and sustainable environments [55], which are being suggested using ambient intelligence.

Issues of *security*, *privacy*, and *trust* in WoT promise many opportunities for research innovations [56]. To clearly contrast these three issues with respect to WoT, consider the example of a car on the Web. *Privacy* would involve dealing with issues that arise when the current owner decides to share the car with others on the Web. *Security* would deal with issues that pertain to who or what will have access to the car when it is in use. *Trust* would deal with issues of interactions between things on the Web, like if the garage door would open when the car arrives. The use of REST-based interfaces makes it possible to have secure interactions using HTTP authentication or HTTPS [57]. As things on the Web will be accessible and shared among many users, research in this area is crucial to the success and widespread use of the WoT. Research has been rampant with innovative ideas dealing with these issues in the IoT [58, 59], but the focus on WoT is yet to mature. The use of the social Web as a platform to ensure the trust and privacy of things has been advocated [60], to control Web-enabled things among trusted members on social Web sites.

7 Conclusion

The Web of Things (WoT) is complementing the Interne of things (IoT) and rapidly expanding as an important area of research. Developments in WoT research have the potential to realize Weiser's vision. Many salient technologies are driving the success of the WoT, such as embedded systems, Web services, IPV6 and tagged things. In this chapter, we have presented the state of the art in WoT research and discussed existing technologies and platforms that support the WoT. We presented different approaches to Web-enable things and facilitate the realization of WoT based applications. The approaches are motivated by the level of comfort that people have when dealing with the Web, to portray the WoT as a simple and do-it-yourself technology. Advances in communication and real-time systems indicate that real-world things will soon become IP-enabled with embed Web servers, making them addressable, controllable and composable on the Web. In this context, we surveyed ongoing research, which addresses some of the challenges posed in leveraging and adapting existing technologies for pushing and pulling information from real-world things. Many innovative ideas and applications are being designed and deployed on WoT to create ambient spaces where people and real-world things seamlessly interact over the Web. We discussed two such case studies to illustrate the developments to realize ambient spaces. Future research directions, trends and challenges were also presented, which are presently driving the WoT into new horizons.

References

[1] Weiser, M.: The Computer for the 21st Century. Scientific American 265(3), 94–104 (1991)

[2] Mulligan, G.: The Internet of Things: Here Now and Coming Soon. IEEE Internet Computing 14(1), 35–36 (2010)

[3] Berners-Lee, T.: The Web of Things, Special theme on the future web, ERCIM News – the European Research Consortium for Informatics and Mathematics (2008), http://ercim-news.ercim.eu/en72/keynote (retrieved on February 12, 2012)

[4] Raggett, D.: The Web of Things: Extending the Web into the Real World. In: van Leeuwen, J., Muscholl, A., Peleg, D., Pokorný, J., Rumpe, B. (eds.) SOFSEM 2010. LNCS, vol. 5901, pp. 96–107. Springer, Heidelberg (2010)

[5] Ostermaier, B., Kovatsch, M., Santini, S.: Connecting Things to the Web using Programmable Low-power WiFi Modules. In: The Proceedings of the 2nd International Workshop on the Web of Things, WoT 2011 (2011)

[6] Guinard, D., Trifa, V.: Towards the Web of Things: Web Mashups for Embedded Devices. In: The Proceedings of the Workshop Mashups, Enterprise Mashups and Lightweight Composition on the Web, MEM 2009 (2009)

[7] Song, Z., Cardenas, A.A., Masuoka, R.: Semantic Middleware for the Internet of Things. In: The Proceedings of the IEEE Internet of Things (IOT), pp. 1–8. IEEE Press (2010)

[8] Cheshire, S., Steinberg, D.H.: Zero Configuration Networking, the Definitive Guide. O'Reilly (2005)

[9] Duquennoy, S., Grimaud, G., Vandewalle, J.J.: Smews: Smart and Mobile Embedded Web Server. In: International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009), pp. 571–576 (2009)

[10] Song, H., Kim, D., Lee, K., Sung, J.: UPnP-Based Sensor Network Management Architecture. In: The Proceedings of the International Conference on Mobile Computing and Ubiquitous Networking (2005)

[11] Allard, J., Chinta, V., Gundala, S., Richard, G.: Jini meets UPnP: An architecture for Jini/UPnP interoperability. In: The Proceedings of the Application and the Internet (2003)

[12] W3C Working Group, Web Services Architecture, `http://www.w3.org/tr/ws-arch`

[13] Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. dissertation (2000)

[14] Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. 'big' web services: making the right architectural decision. In: The Proceedings of the 17th International Conference on World Wide Web. WWW 2008, pp. 805–814. ACM, New York (2008)

[15] Luckenbach, T., Gober, P., Arbanowski, S., Kotsopoulos, A., Kim, K.: TinyREST: A protocol for integrating sensor networks into the Internet. In: The Proceedings of the REALWSN 2005 (2005)

[16] Drytkiewicz, W., Radusch, I., Arbanowski, S., Popescu-Zeletin, R.: pREST: a REST-based protocol for pervasive systems. In: 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pp. 340–348. IEEE (2004)

[17] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H., Thatte, S., Winer, D.: Simple object access protocol (SOAP) 1.1 (2000)

[18] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web services description language (WSDL) 1.1 (2001)

[19] W3C, Web Services Addressing, `http://www.w3.org/2002/ws/addr/`

[20] Bellwood, T., Capell, S., Clement, L., Colgrave, J., et al.: UDDI Version 3.0.2, Oasis, `http://uddi.org/pubs/uddi-v3.0.2-20041019.htm`

[21] OASIS. Web Services Resources Framework, WSRF 1.2 (2006), `http://www.oasis-open.org/committees/wsrf/`

[22] Weerawarana, S., Curbera, F., Leymann, F., Storey, T., Ferguson, D.: Web Services Platform Architecture. Prentice Hall (2005)

[23] de Deugd, S., Carroll, R., Kelly, K., Millett, B., Ricker, J.: SODA: Service oriented device architecture. IEEE Pervasive Computing 5(3), 94–96 (2006)

[24] Pintus, A., Carboni, D., Piras, A., Giordano, A.: Connecting Smart Things through Web Services Orchestrations. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 431–441. Springer, Heidelberg (2010)

[25] Jammes, F., Smit, H.: Service-oriented paradigms in industrial automation. IEEE Transactions on Industrial Informatics 1(1), 62–70 (2005)

[26] Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., de Souza, L.M.S., Trifa, V.: SOA-based integration of the internet of things in enterprise services. In: The Proceedings of the 2009 IEEE International Conference on Web Services (ICWS 2009), pp. 968–975. IEEE Computer Society (2009)

[27] Guinard, D., Mueller, M., Trifa, V.: RESTifying Real-World Systems: a Practical Case Study. In: RFID. Springer (2011)

[28] Guinard, D.: A Web of Things Application Architecture - Integrating the Real-World into the Web. Ph.D. dissertation (2011)

[29] Duquennoy, S., Grimaud, G., Vandewalle, J.J.: The Web of Things: Interconnecting Devices with High Usability and Performance. In: The Proceedings of the International Conference on Embedded Software and Systems, ICESS 2009 (2009)

[30] Agranat, I.: Engineering web technologies for embedded applications. IEEE Internet Computing 2(3), 40–45 (1998)

[31] The Web of Things, White Paper, Tridium Inc. (September 2009)

[32] Lizcano, D., Jiménez, M., Soriano, J., Cantera, J.M., Reyes, M., Hierro, J.J., Garijo, F., Tsouroulas, N.: Leveraging the Upcoming Internet of Services through an Open User-Service Front-End Framework. In: Mähönen, P., Pohl, K., Priol, T. (eds.) ServiceWave 2008. LNCS, vol. 5377, pp. 147–158. Springer, Heidelberg (2008)

[33] Filibeli, M.C., Ozkasap, O., Reha Civanlar, M.: Embedded Web server-based home appliance networks. Journal of Network and Computer Applications 30, 499–514 (2007)

[34] Priyantha, N.B., Kansal, A., Goraczko, M., Zhao, F.: Tiny web services: design and implementation of interoperable and evolvable sensor networks. In: The Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. SenSys 2008, pp. 253–266. ACM, New York (2008)

[35] Kortuem, G., Kawsar, F., Sundramoorthy, V., Fitton, D.: Smart objects as building blocks for the internet of things. IEEE Internet Computing, 30–37 (January/February 2010)

[36] Mathew, S.S.: Managing things in an Ambient Space. In: The Proceedings of the 9th International Conference on Service Oriented Computing (ICSOC), PhD Symposium, Paphos, Cyprus (2011)

[37] Zeng, D., Guo, S., Cheng, Z.: The Web of Things: A Survey. Journal of Communications 6(6), 424–438 (2011)

[38] Akribopoulos, O., Chatzigiannakis, I., Koninis, C., Theodoridis, E.: A Web Services oriented Architecture for Integrating Small Programmable Objects in the Web of Things. Developments in E-systems Engineering, 70–75 (2010)

[39] Trifa, V., Wiel, S., Guinard, D., Bohnert, T.: Design and implementation of a gateway for web-based interaction and management of embedded devices. In: The Proceedings of the 2nd International Workshop on Sensor Network Engineering, IWSNE (2009)

[40] Guinard, D., Floerkemeier, C., Sarma, S.: Cloud computing, rest and mashups to simplify RFID application development and deployment. In: The Proceedings of the 2nd International Workshop on the Web of Things (WoT 2011). ACM, San Fransisco (2011)

[41] Welbourne, E., et al.: Building the Internet of Things Using RFID: The RFID Ecosystem Experience. IEEE Internet Computing (May 2009)

[42] Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., Savio, D.: Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. IEEE Transactions on Services Computing 3, 223–235 (2010)

[43] Guinard, D.: Mashing Up Your Web-Enabled Home. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 442–446. Springer, Heidelberg (2010)

[44] Garrett, J.J.: Ajax: A new approach to Web applications. Adaptivepath (2005)

[45] Russell, A.: Comet: Low Latency Data For Browsers. The Dojo Toolkit (2006)

[46] Campus Conservation Nationals: Competition to Slash Energy Use at US Colleges, Alliance to Save Energy (November 10, 2011), http://ase.org/efficiencynews/campus-competition-slash-energy-use

[47] Ostermaier, B., Romer, K., Mattern, F., Fahrmair, M., Kellerer, W.: A Real-Time Search Engine for the Web of Things. In: The Proceedings of the International Conference on Internet of Things (IoT 2010), Japan (2010)

[48] Mayer, S., Guinard, D.: An extensible discovery service for smart things. In: The Proceedings of the 2nd International Workshop on the Web of Things (WoT 2011), USA. ACM (2011)

[49] Mathew, S.S., Atif, Y., Sheng, Q.Z., Maamar, Z.: Ambient things on the Web. Journal of Ubiquitous Systems and Pervasive Networks (JUSPN) 1(1), 1–8 (2010)

[50] Shelby, Z.: Embedded Web services. IEEE Wireless Communications 17(6), 52–57 (2010)

[51] Mulligan, G.: The 6LoWPAN architecture. In: The Proceedings of the Fourth Workshop on Embedded Networked Sensors (EmNets 2007), Ireland, pp. 78–82. ACM (2007)

[52] Boussard, M., Christophe, B., Le Berre, O., Toubiana, V.: Providing user support in Web-of-Things enabled Smart Spaces. In: The Proceedings of the Second International Workshop on Web of Things (WoT 2011). ACM, USA (2011)

[53] Compagna, L., El Khoury, P., Massacci, F., Saidane, A.: A Dynamic Security Framework for Ambient Intelligent Systems: A Smart-Home Based eHealth Application. In: Gavrilova, M.L., Tan, C.J.K., Moreno, E.D. (eds.) Transactions on Computational Science X. LNCS, vol. 6340, pp. 1–24. Springer, Heidelberg (2010)

[54] Mantoro, T., Ayu, M.A., Elnour, E.E.: Web-enabled smart home using wireless node infrastructure. In: The Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2011). ACM, USA (2011)

[55] Paetz, A., Dütschke, E., Fichtner, W.: Smart Homes as a Means to Sustainable Energy Consumption: A Study of Consumer Perceptions, From Knowledge to Action - New Paths towards Sustainable Consumption. Journal of Consumer Policy (2012)

[56] Gupta, V., Udupi, P., Poursohi, A.: Early lessons from building Sensor.Network: an open data exchange for the web of things. In: The Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops (2010)

[57] Wilde, E.: Putting Things to REST, UCB iSchool Report 2007-015, School of Information, UC Berkeley (2007)

[58] Weber, R.H.: Internet of Things - New security and privacy challenges. Computer Law & Security Review 26(1) (2010)

[59] Medaglia, C.M., Serbanati, A.: An overview of privacy and security issues in the internet of things. In: The Internet of Things, pp. 389–395. Springer, New York (2010)

[60] Guinard, D., Fischer, M., Trifa, V.: Sharing Using Social Networks in a Composable Web of Things. In: The Proceedings of the 1st IEEE International Workshop on the Web of Things (WoT 2010) at IEEE PerCom, Germany (2010)

[61] Wu, Y., Sheng, Q.Z., Ranasinghe, D., Yao, L.: PeerTrack: A Platform for Tracking and Tracing Objects in Large-Scale Traceability Networks. In: The Proceedings of the 15th International Conference on Extending Database Technology (EDBT 2012), Germany (2012)

Context-Aware Environments for the Internet of Things

Valentin Cristea, Ciprian Dobre, and Florin Pop

Abstract. This chapter discusses the challenges, state of the art, and future trends in context aware environments (infrastructure and services) for the Internet of Things, which is defined as a world-wide network of uniquely identified self-organized and intelligent things. Intelligence means primarily the capability of things to be aware of the context in which they operate (time, geographic location, geographic dimension, situation, etc.) and to inter-cooperate with other things in the environment. The Chapter is structured in three sections. The first section, which frames the issues discussed in the rest of the chapter, is a systematic presentation of the most relevant concepts and aspects related to the infrastructure and services for the Internet of Things. The second section presents relevant research works in the infrastructure, and up to date solutions and results regarding the infrastructure and services. The third section presents future trends and research directions in the domain.

Keywords: Context-Aware Environments, Services, Collective Intelligence, Scalability, High-Performance, Internet of Things.

1 Introduction to Internet of Things Infrastructure and Services

This section, which frames the issues discussed in the rest of the chapter, is a systematic presentation of the most relevant concepts, aspects and main issues related to the context-aware infrastructure and services for the Internet of Things (IoT). For the purpose of this chapter we adopt the IoT definition presented in [5]:

> *"The Internet of Things (IoT) is an integrated part of the Future Internet and could be defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network. In the IoT, "things" are expected to become active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information "sensed" about the*

Valentin Cristea · Ciprian Dobre · Florin Pop
University *Politehnica* of Bucharest, Computer Science Department
Splaiul Independentei 313, Bucharest 060042, Romania
e-mail: {valentin.cristea,ciprian.dobre,florin.pop}@cs.pub.ro

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 25–49.
DOI: 10.1007/978-3-642-34952-2_2

> *environment, while reacting autonomously to the "real/physical world" events and influencing it by running processes that trigger actions and create services with or without direct human intervention. Interfaces in the form of services facilitate interactions with these "smart things" over the Internet, query and change their state and any information associated with them, taking into account security and privacy issues."*

As part of the Future Internet, IoT aims to integrate, collect information from-, and offer services to a very diverse spectrum of physical things used in different domains. "Things" are everyday objects for which IoT offers a virtual presence on the Internet, allocates a specific identity and virtual address, and adds capabilities to self-organize and communicate with other things without human intervention. To ensure a high quality of services, additional capabilities can be included such as context awareness, autonomy, and reactivity.

This section starts with an introductory presentation of things and IoT infrastructure and continues with the main functional aspects related to things' intercommunication, the context model for the IoT and the event-driven mechanisms to sense, process, and exchange context data. Non-functional requirements for the IoT infrastructure are described in the end.

Things are very diverse. Simple things, like books, can have Radio Frequency Identification - RFID tags that help tracking them without human intervention. For example, in an electronic commerce system, a RFID sensor network can detect when a thing leaves the warehouse and can trigger specific actions like inventory update or customer rewarding for buying a high end product [1]. In this simple case, RFIDs enable the automatic identification of things, the capture of their context (for example the location) and the execution of corresponding actions if necessary. Sensors and actuators are used to transform real things into *virtual objects* [3] [5] with digital identities. In this way, things may communicate, interfere and collaborate with each other over the Internet [6]. Adding part of application logic to things transforms them into *smart objects* [15], which have additional capabilities to sense, log and understand the events occurring in the physical environment, autonomously react to context changes, and intercommunicate with other things and people. A tool endowed with such capabilities could register when and how the workers used it and produce a financial cost figure. Similarly, smart objects used in the e-health domain could continuously monitor the status of a patient and adapt the therapy according to monitoring results. Smart objects can also be general purpose portable devices like smart phones and tablets, that have processing and storage capabilities, and are endowed with different types of sensors for time, position, temperature, etc. Both specialized and general purpose smart objects have the capability to interact with people.

The IoT includes a hardware, software and services infrastructure for things networking. IoT infrastructure is event-driven and real-time, supporting the context sensing, processing, and exchange with other things and the environment. The infrastructure is very complex due to the huge number (50 to 100 trillion) of heterogeneous, (possibly) mobile things that dynamically join and leave the IoT, generate and consume billions of parallel and simultaneous events geographically distributed all over the world. The complexity is augmented by the difficulty to represent, interpret, process, and predict the diversity of possible contexts. The infrastructure must have important characteristics such as reliability, safety, survivability, security and fault tolerance. Also, it must manage the communication, storage and compute resources.

The IoT infrastructure supports communication among things. This function must be flexible and adapted to the large variety of things, from simple sensors to sophisticated smart objects. More specific, things need a communication infrastructure that is low-data-rate, low power, and low-complexity. Actual solutions are based on short-range radio frequency (RF) transmissions in ad-hoc wireless personal area networks (WPANs). A main concern of IoT infrastructure developers is supporting heterogeneous things [42] by adopting appropriate standards for the physical and media access control (MAC) layers, and for communication protocols. The protocol and compatible interconnection for the simple wireless connectivity with relaxed throughput (2 – 250 kb/s), low range (up to 100 m), moderate latency (10 – 50 ms) requirements and low cost, adapted to devices previously not connected to the Internet were defined in IEEE 802.15.4 [7]. Other similar efforts refer to industrial and vehicular applications - ZigBee [8], open standards for process control in industrial automation and related applications - ISA100.11a [9] and WirelessHART [10], and encapsulating IPv6 datagrams in 802.15.4 frames, neighbor discovery and routing that allow sensors to communicate with Internet hosts - 6LoWPAN [11]. The scope of IoT specialists is the worldwide network of interconnected virtual objects uniquely addressable and communicating through standard protocols.

The IoT architecture supports physical things' integration in Internet and the complex interaction flow of services triggered by event occurrences. Objects identification, sensing and connecting capabilities form the basis for the development of independent cooperative services and applications that address key features of the IoT architecture: Service Orientation, Web-base, distributed processing, easy integration via native XML and SOAP messaging, component-base, open access, N-tiered architecture, support for vertical and horizontal scalability [13]. Specific Web services help the physical objects to “become active participants in business processes” [14]. They interact with the corresponding virtual objects over the Internet, query and change objects’ state, and process other associated information. The new key features for the IoT architecture include persistent messaging for the highest availability, complete security and reliability for total control and compliance, platform independence and interoperability (more specific for middleware).

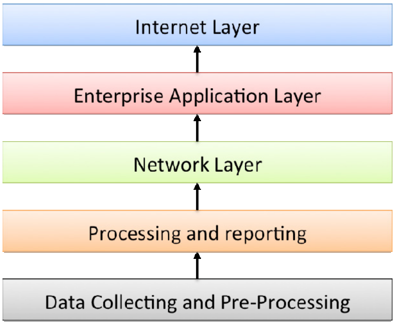

Internet Layer
Enterprise Application Layer
Network Layer
Processing and reporting
Data Collecting and Pre-Processing

Fig. 1 Layered Networking for IoT

IoT infrastructure considers extended process functionality, pathways and layered networks as main components. These layers (see Fig. 1) refer to: real-time Data Collecting and Pre-Processing, which aims to support any available or emerging technology and offer uniform interfaces to upper layers; Processing and reporting environment data according to specific rules and data packing; efficient data transportation over the Network; support for Enterprise Application development; and application exposure and integration in the Internet. The IoT infrastructure supports object-connected technologies for "Human-to-Objects" and "Objects-to-Objects" communications [2] [4]. The communication platforms are heterogeneous, ad-hoc, and opportunistic.

As mentioned previously, IoT is a large heterogeneous collection of things, which differ from each other. Even things that have the same nature, construction, and properties can differ from one another by their situation or context. Context means the conditions in which things exist in other words their surrounding world. Since virtual things in IoT are interconnected, the meaning of the data they exchange with other things and people becomes clear only when it is interpreted in the thing's context. This is why the IoT infrastructure runs reliably and permanently to provide the context as a "public utility" to IoT services [31]. For human users, the context is the information that characterizes user's interaction with Internet applications plus the location where this interaction occurs, so that the service can be adapted easily to users' preferences, For things, we need another approach. A very suggestive example is given in [33]. The authors explain the case of a plant that is the target of an automatic watering service. In order to control the watering dosages and frequency, the service has to sense the dryness status of the plant, to use the domain knowledge of plants and find their watering "preferences", and to ask the weather prediction service about the chances of rain in the next days. So, the context of a thing includes information about thing's environment and about the thing itself [33].

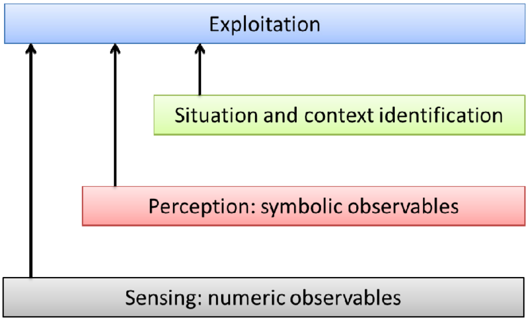

Exploitation
Situation and context identification
Perception: symbolic observables
Sensing: numeric observables

Fig. 2 Context-aware services

Several context modeling and reasoning techniques are known today [34], some of them being based on knowledge representation and description logics. Ontology-based models can describe complex context data, allow context integration among several sources, and can use reasoning tools to recognize more abstract contexts. Ontologies provide a formal specification of the semantics of context data that stay at the base of knowledge sharing among different things in IoT. In addition, ontological reasoning can derive new context. Ontology-based models can be used to organize IoT infrastructure context-aware services as a fabric structured into multiple levels of abstraction (see Fig. 2) starting with collecting information from physical sensors (called low level context), which could be meaningless and consequently not useful to applications. Next, higher-level context is derived by reasoning and interpretation. Finally, context is exploited by triggering specific actions [31].

The IoT infrastructure combines the context model with event-based organization of services that support the collection, transmission, processing, storage and delivery of context information. In the event-driven architecture vocabulary, events are generated by different sources, event producers, when for example a context change is significant and must be propagated to target applications, which are event consumers. Producers and consumers are loosely coupled by the asynchronous transmission and reception of events. They don't have to know and explicitly refer each other. In addition, the producers don't know if the transmitted events are consumed ever. A publish/subscribe mechanism is used to offer the events to the interested consumers. Other components are the event channels for communication, and event processing engines for complex event detection. Components for event specification, event management, and for the integration of the event-driven system with application belong also to the IoT infrastructure.

There are also non-functional requirements associated with IoT infrastructure [1]: large scale integration, interoperability between heterogeneous things, fault tolerance and network disconnections, mobility, energy saving, reliability, safety, survivability, protection of users' personal information (e.g., location and preferences) against security attacks, QoS and overall performance, scalability, self-* properties and transparency.

Development issues for IoT infrastructure are directly related to Service Oriented Architecture (SOA), Collaborative Decision Making (CDM), Cloud Computing, Web 2.0 (and Future Internet) and Semantic Web. The support of 6A connectivity (Anything, Anyone, Anytime, Any Place, Any Service, and Any Network) becomes the most important key feature for adding sense to the Internet of Things [13].

2 Context Aware Internet of Things Infrastructure

This section presents up to date solutions and research results regarding the structure, characteristics, and services for context aware IoT infrastructure.

Sensors in IoT are used to collect various data such as biomedical information, environment temperature, humidity, and ambient noise level. The data provided by such sensors can be used by customized context-aware applications and services, capable to adapting their behavior to their running environment. However, sensor data exhibits high complexity (e.g., due to the huge volumes and interdependency relationships between sources), dynamism (e.g., updates performed in real-time and data that can age until it becomes useless), accuracy, precision and timeliness. An IoT system should not concern itself with the individual pieces of sensor data: rather, the information should be interpreted into a higher, domain-relevant concept. For example, sensors might monitor temperature, humidity, while the information needed by a watering actuator might be that the environment is dry. This higher-level concept is called a situation, which is an abstract state of affairs interesting to applications [19].

2.1 Situational Awareness

Situations are generally representations (simple, human understandable) of sensor data. They shield the applications from the complexities of sensor readings, sensor data noise and inferences activities. However, in large-scale systems there may be tens or hundreds of situations that applications need to recognize and respond to. Underlying these situations will be an even greater number of sensors that are used in situation identification. A system has a significant task of defining and managing these situations. This includes capturing what and how situations are to be recognized from which pieces of contexts, and how different situations are related to each other. The system should know, for example, which situations can or cannot occur: a room hosting a “scientific event” and an “academic class” at the same time); otherwise, inappropriate adaptive behavior may occur. Temporal order between situations is also important, such as the inability of a car to go directly from a situation of ‘parked’ to ‘driving on a highway’. Given the inherent inaccuracy of sensor data and the limitations of inference rules, the detection of situations is imperfect.

The research topics on situation identification for IoT involve several issues [20]. First, *representation* deals with defining logic primitives used to construct a situation’s logical specification. In representation, logical primitives should capture features in complex sensor data (e.g., acceleration data), scope knowledge (e.g., a spatial map or social network), and different relationships between situations. Also, an IoT system is assumed to be highly dynamic. New sensors can be introduced, that bring new types of context. Therefore, the logical primitives should be flexibly extensible, such as new primitives to not cause modifications or produce ambiguous meanings to existing ones.

Specification deals with defining the logic behind a particular situation. This can be acquired by experts or learned from training data. It typically relies on a situation model with a priori expert knowledge, on which reasoning is applied based on the input sensor data. For example, in logic programming [37] the key underlying assumption is that knowledge about situations can be modularized or digitized. An example adapted from [21], which defines a situation when a room is detected as being occupied, can be specified as follows:

```
Occupied(room) = ∃t1, t2, event|reservedBy(person, t1, t2)
        •((t1 ≤ timenow() ∧ (timenow() ≤ t2 ∨ isnull(t2)))
        ∨((t1 ≤ timenow() ∨ isnull(t1)) ∧ timenow() ≤ t2))
```

These initial works have been advanced to a more formal approach by Loke et al. [22]. The authors proposed a declarative approach to represent and reason with situations at a higher level of abstraction. The approach uses logical programming in Prolog to embed situations. For example a situation `in_meeting_now` of a user entity E is defined on two situations `with_someone_now` and `has_entry_for_meeting_in_diary` can be defined as:

```
if in_meeting_now(E) then
        with_someone_now(E),
        has_entry_for_meeting_in_diary(E).
if with_someone_now(E) then
        location*(E, L), people_in_room*(L, N) , N > 1.
if has_entry_for_meeting_in_diary(E) then
        current_time*(T1) ,
        diary*(E, ‘meeting’, entry(StartTime, Duration)),
        within_interval(T1, StartTime, Duration).
```

Each of these situations is defined on sensor predicates. For example, `with_someone_now` refers to two sensor predicates: `location*(E, L)` that returns the location of the entity, and `people_in_room*(L, N)` that returns the number of people in the location. These predicates refer to lower-level context, as detected by various sensors. In this way, situation programs can be made amenable to formal analysis, and the inference procedure of reasoning about situations is decoupled from the acquisition procedure of sensor readings.

2.2 Other Representation Approaches

Other logic theories, such as situation calculus [38], have also been used to infer situations in IoT systems. Kalyan et al. [23] introduce a multi-level situation theory, where an intermediate level micro situation is introduced between infons and situations. An infon embodies a discrete unit of information for a single entity (e.g., a customer or a product), while a situation makes certain infons factual and thus supports facts. Micro situations are composed of these entity-specific infons, which can be explicitly obtained from queries or implicitly derived from sensors and reasons. Situations are considered as a hierarchical aggregation of micro situations and situations. This work aims to assist information reuse and support ease of retrieving the right kind of information by providing appropriate abstraction of information.

Spatial and temporal logic has also been applied to represent and reason on spatial and temporal features and constraints of context and situations. Augusto et al. [24] introduce the temporal operators ANDlater and ANDsim in Event–Condition–Action rules, upon which temporal knowledge on human activities can be specified. Considering the sensor events `at_kitchen_on` (the activation of the RFID sensors in the kitchen), `tkRK_on` (the activation of the RFID sensor while the user is passing through the door between the kitchen and the reception area), and `no_movement_detected` (sensing of no movement), the following rule specifies a situation of a user ‘fainting’:

```
IF at_kitchen_on ANDlater tdRK_on ANDlater no_movement_detected
THEN assume the occupant has fainted
```

Also, ontologies have increasingly gained attention as a generic, formal and explicit way to capture and specify the domain knowledge with its intrinsic semantics through consensual terminology and formal axioms and constraints. They provide a formal way to represent sensor data, context, and situations into well-structured terminology. Based on the modeled concepts, developers can define logical specifications of situations in rules. An exemplar rule on an activity ‘sleeping’ is given in [25]:

```
(?user rdf:type socam:Person),
(?user, socam:locatedIn, socam:Bedroom),
(?user, socam:hasPosture, ‘LIEDOWN’),
(socam:Bedroom, socam:lightLevel, ‘LOW’),
(socam:Bedroom, socam:doorStatus, ‘CLOSED’)
-> (?user socam:status ‘SLEEPING’)
```

Ontologies, together with their support for representation formalisms, can support reasoning, including detecting inconsistency, or deriving new knowledge. An ontological reasoner can be used to check consistency in a class hierarchy and consistency between instances, e.g. whether a class is being a subclass of two classes that are declared as disjoint or whether two instances are contradictory to each other (such as a person being detected in two spatially disjoint locations at the same time). Given the current sensor data, the reasoner will derive a new set of statements. In the above ‘sleeping’ example, if the reasoner is based on a forward-chaining rule engine, it can match the conditions of this rule against the sensor input. If all the conditions are satisfied, the reasoner will infer the conclusion of the rule. The reasoning will terminate if the status of the user is inferred, when the status of the user is set to be the default inference goal in this reasoner.

Other solutions are based on the Dempster–Shafer theory (DST) [39], a mathematical theory of evidence, which propagates uncertainty values and consequently provides an indication of the certainty of inferences. The process of using DST is described as follows. First, developers apply expert knowledge to construct an evidential network that describes how sensors lead to activities. The left-hand side of Fig. 3 below describes that the sensors on the cup and fridge are connected to context information (e.g., ‘cup used’). Such context information can be further inferred or composed to higher-level context. The composition of context information points to an activity (e.g., ‘Get drink’) at the top. Developers can use such an approach to determine the evidence space and degree of belief in an evidence. For example, in Fig. 3, the values on the arrows represent the belief in particular sensor (also called the uncertainty of sensor observations). Generally, in reasoning situations are inferred from a large amount of imperfect sensor data. In reasoning, one of the main processes is called situation identification - deriving a situation by interpreting or fusing several pieces of context in some way. Specifying and identifying situations can have a large variability depending on factors such as time, location, individual users, and working environments [26]. This makes specification-based approaches relying on models of a priori knowledge impractical to use. Machine learning techniques have been widely applied to learning complex associations between situations and sensor data. However, the performance of reasoning is usually undermined by the complexity of the underlying sensor data.

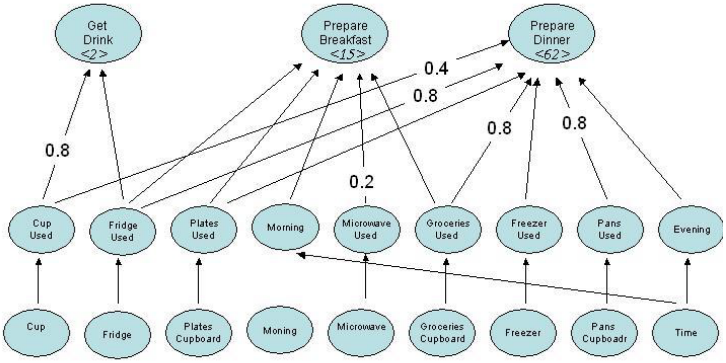


Fig. 3 An example of situation inferring using Dempster-Shafer theory (from [40])

Bayesian networks have been applied in many context-aware systems. For example, in [27] the authors present a solution to infer a user's current activity from sensors within a room, which provide several contexts (e.g., several sensors track people within a house, the light level and noise level in rooms are monitored by other sensors). But, such a Bayesian network cannot model the causal relationship between the status context of one particular user, his/her location, and the status of the micro oven for example, because of the breaking of the independence assumption in naïve Bayes. A better approach consists in the use Hidden Markov Models (HMMs) [28]. HMMs statistical models where a system being modeled is assumed to be a Markov chain that is a sequence of events. A HMM is composed of a finite set of hidden states and observations that are generated from states. For example, a HMM where each state represents a single activity (e.g., 'prepare dinner', 'go to bed', 'take shower', and 'leave house') is presented in [28]. They represent observations in three types of characterized sensor data that are generated in each activity, which are raw sensor data, the change of sensor data, the last observed sensor data, and the combination of them. The HMM is trained to obtain the three probability parameters, where the prior probability of an activity represents the likelihood of the user starting from this activity; the state transition probabilities represent the likelihood of the user changing from one activity to another; and the observation emission probabilities represent the likelihood of the occurrence of a sensor observation when the user is conducting a certain activity.

Unlike this approach, [29] employs the use of neural networks in learning activities (e.g., static activities like 'sitting' and 'working at a PC', and dynamic activities like 'running' and 'vacuuming') from acceleration data. A similar idea was further applied to detect bump holes as cars runs on street, using accelerometer sensors inside the vehicle [41]. The acceleration data is collected from a wireless sensing tri-axial accelerometer module, from which eight features are extracted, including the mean value, the correlation between axes, and the energy that is used to discriminate between activities.

Finally, Support Vector Machines (SVM) [12] is a relatively new method for classifying both linear and nonlinear data. An SVM uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyper-plane that separates the training data of one class from another. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated. SVMs are good at handling large feature spaces since they employ over fitting protection, which does not necessarily depend on the number of features. Kanda et al [30] use SVMs to categorise motion trajectories (such as ‘fast’, ‘idle’, and ‘stop’) based on the velocity, direction, and shape features extracted from various sensors (within a car for example). Different types of sensor data lead to different techniques to analyze them. Numerical data, for example, can be used to infer motions like ‘walking’ or ‘running’ from acceleration data. Situation identification at this level is usually performed in learning-based approaches, which uncover complicated associations (e.g., nonlinear) between continuous numerical data and situations by carving up ranges of numerical data (e.g., decision tree) or finding an appropriate algebraic function to satisfy or ‘explain’ data (e.g., neural networks or SVMs). Specification-based approaches can apply if the association between sensor data and situations are rather explicit and can be represented in logic rules. Situations can also be recognized from categorical features; for example, inferring a room’s situation - ‘meeting’ or ‘presentation’ — from the number of persons co-located in the room and the applications running in the computer installed in the room. This higher-level of situation identification can be performed using either specification- or learning-based approaches.

Uncertainty can also exist in the use of oversimplified rules that are defined in an ad hoc way. In representing uncertainty of rules, Web Ontology Language (OWL), a family of knowledge representation languages for authoring ontologies endorsed by W3C, can be extended with a conditional probabilistic class to encode the probability that an instance belongs to a class respectively given that it belongs to another class. Although good at expressing uncertainty, these qualitative approaches need to be combined with other machine-learning techniques to quantify the uncertainty to be used in situation identification. Learning-based approaches have a stronger capability to resolve uncertainty by training with the real-world data that involves noise. These approaches not only learn associations between sensor data and situations, but also the effect that the uncertainty of sensor data has on the associations. For example, the conditional probabilities learned in Bayes networks include the reliability of sensor data as well as the contribution of the characterized sensor data in identifying a situation.

2.3 Architectural Issues

A popular architectural model for IoT is composed of autonomous physical/ digital objects augmented with sensing, processing, and network capabilities.

Unlike RFID tags, smart objects carry an application logic that let them sense their local situation and interact with the environment through actuators. They sense, log, and interpret what's occurring within themselves and the work, act on their own, intercommunicate with each other, and exchange data [17].

According to the scenarios illustrated in [17], the architectural differences in the way smart objects understand (sense, interpret or react to) events, and interact with their environment in terms of input, output, control and feedback, classify them as either activity-aware objects, policy-aware objects or process-aware objects. A process-aware object represents the most accomplished type, and characterizes: awareness (a process-aware object understands the operational process that is part of and can relate the occurrence of real-work activities and events to these processes), representation (its model consists of a context-aware workflow model that defines timing and ordering or work activities), and interaction (a process-aware object providers workers with context-aware guidance about tasks, deadlines, and decisions).

Adaptable Pervasive Flows [18] is a technology that model applications in a fashion similar to classical service workflows, while being situated in the real world. A flow is a computer-based model that essentially consists of a set of actions, glued together by a plan (or control flow) which defines how the actions should be performed to achieve some goal under a set of constraints. Flows are explicitly tailored (1) to being executed in pervasive environments, and (2) to being adaptable. They are situated in the real world, i.e., they are logically attached to entities, they can move with them through different contexts. While they are carried along, they model the behavior intended for the associated entity, and adapt the entity's environment to this behavior. Thus, when a mobile user carries a flow that specifies his prospective actions, the pervasive computing machinery in his environment will be set up for him by the flow. Since people may change their minds, and since artifacts and people may be subject to changes in the environment, the flow itself may also adapt to reflect such changes. This requires flows to be context-aware. They can take into account the context pertaining to their entity's current environment as well as the entity's actual activities in order to dynamically adapt to changing situations.

A context-aware infrastructure designed to support smart objects could help workers in construction industry by providing just-in-time information about required work activities [17]. To model the organizational process a workflow [18] can be used to define work activities in which the smart object is involved. Such a workflow can contain activities and transitions between activities. Transitions can be annotated with context conditions that refer to sensor or human input. A workflow continues along a transition if input satisfies a condition.

The goal of JCAF [16] is to create a general-purpose, robust, event-based, service-oriented infrastructure and a generic, expressive Java programming framework for the deployment and development of context-aware IoT applications. The infrastructure is composed of a Context-awareness Runtime Infrastructure and a Context-awareness Programming Framework. Each Context Service is a long-lived process analog to a J2EE Application Server. The service's Entity Container manages an Entity with its Context information. An entity is a small Java program that runs within the Context Service and responds to changes in its context. The life cycle of an entity is controlled by the container in which the entity has been added. The entity container handles subscribers to context events and notifies relevant clients on changes to entities.

2.4 CAPIM Infrastructure

An infrastructure that follows these concepts and the hierarchical view of functions is presented in CAPIM [2], a platform for context aware IoT systems [31] that integrates smartphones for students and staff interactions in a university campus. The platform collects and manages a global context (of the surrounding space) by integrating capabilities of various sensors and actuators. Such sensors are aggregated using the sensing, processing and communication capabilities of smart objects, in particular smartphones. Smart objects can support the integration in Internet of parking lots, university restaurants, libraries, classrooms, administrative offices, etc. and can communicate with each other and with smartphones for their exploitation in collaborative e-services dedicated to students and staff. Since smartphones become commodity hardware, used almost everywhere, having more sensing and computing capabilities in every-day situations is attractive for many reasons. Smartphones can sustain next-generation efforts of making the Internet of Things vision a reality – users and devices blend together smoothly in a single virtual world where smartphone, coupled with other sensors and services from the environment, can optimize (e.g. by helping organizing tasks, contacts, etc.) and assist (e.g. with navigation, find information more quickly, access online data, etc.) users in everyday activities. These may refer to finding a vacant parking space in a parking lot that is closer to user's office, classroom or actual car position, assisting people parking and pay for parking, finding the best way towards a specific classroom, getting information about the activity in that room or about the number of people who are actually there, being notified about new publications available in the library or about the actual menu of the preferred restaurant, etc. This is a shift towards developing mobile context-aware services that are capable to recognize and pro-actively react to user's own environment. Such context-aware mobile applications can help things better interact between themselves and with their surrounding environments, and offer high quality information to people. This is the basis for a paradigm where the context is actively used by applications designed to take smarter and automated decisions: start the cooling system when the temperature raises above a specific threshold and there is a meeting in that room, mute the phone of users participating to the meeting, show relevant information for the user's current location, assist the user find its way in the campus, or automatically recommend events based on the user's (possible learned) profile and interests.

CAPIM is designed to support the construction of the next-generation context aware applications. It integrates services designed to collect context data (things' location, profile, etc.). These smart services can be dynamically loaded by mobile things, and make use of the sensing capabilities provided by modern smart objects, including smartphones endowed with additional sensors. The data is collected and aggregated into context instances. This is also possibly augmented with external and inferred data about situations, relations, and events. In addition, the platform includes a workflow engine designed to continuously evaluate the context and take automatically decisions or actions based on customized rules and particular context events.

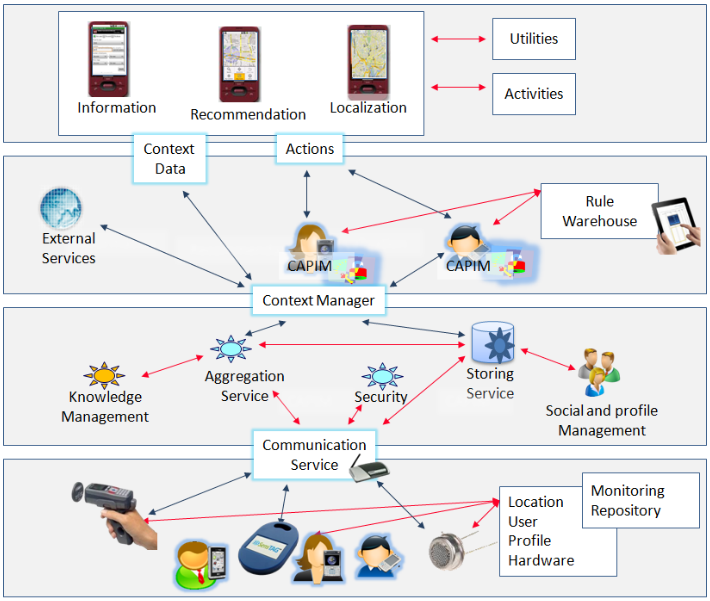


Fig. 4 CAPIMs'architecture

CAPIM's architecture consists of four vertical layers (see Fig. 4). Each layer provides a specific function: (1) collecting context information, (2) storing and aggregation of context information, (3) construction of context-aware execution rules, and (4) visualization and user interaction. Each layer has several components, making the infrastructure suitable for experimenting with a wide range of context-aware things, methods, techniques, algorithms, and technologies. CAPIM can be used to construct context-aware applications using a service-oriented composition approach: load a core container, instruct it to load the necessary context-gathering services, deploy a corresponding context-aware business workflow, and call the actions to be executed when context is met. For example, the monitoring services are dynamically discovered, downloaded as needed, loaded and executed in the container. The first layer includes sets of monitoring services for collecting context data and first-stage storing on the local smart objects.

Each monitoring service is packed in a digitally signed monitoring module. These modules are downloadable from remote repositories, resembling application stores. The monitoring services can be developed/maintained by third party organizations. For example, a manufacturer might build a module to collect data from sensors it is offering on the market, therefore integrating them within Internet.

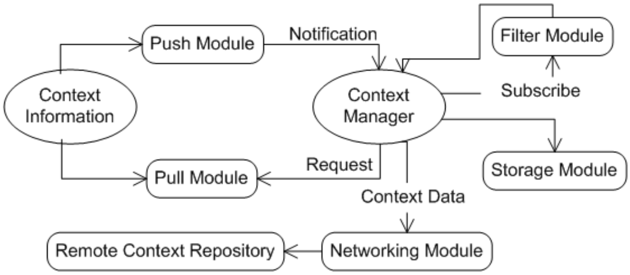


Fig. 5 Flow of monitoring information

Each monitoring service is executed in a separate container. This allows separation of concerns (no service needs to know what other modules are deployed) and fault isolation.

The monitoring flow (see Fig. 5) is under the control of a Context Manager, orchestrating the flow of information between the monitoring services. Depending on the function supported, the monitoring services are grouped in several categories. The Push and Pull monitoring services are directly responsible for collecting context information, usually directly from sensors. The Push service reacts to changes of the context, which in turn triggers notifications to the Context Manager. The Pull service is periodically or on-request interrogated for the current values of the monitoring parameters.

The context information is sent to Filter, Storage and Networking services. The Filter service subscribes to specific context information. The Context Manager forwards the data of interest to the Filter service, which in turns can produce new context information (possible from multiple data sources). Such a construction allows for first-stage aggregation of context information.

The Storage service can keep data locally for better serving the context-execution rules. Finally, the Networking service is responsible for sending the collected context information remotely to aggregation services (the Remote Context Repository component located in the next layer). Here we can experiment with different network protocols and methods of sending data, whilst balancing between costs and energy-consumption.

Each monitoring service is also responsible for a particular type of monitoring information. Thus, these services fall into different categories: location, profile, hardware.

The second layer deals with the aggregation and storing of context data. The components at this layer are running in a server environment, mainly because the aggregation involves collecting data from multiple fixed and mobile sources. Also it involves higher computational capabilities that are available on smart things and user's smartphone without interfering with his/her own activities. The components are distributed, and we envision a scheme where several such servers collect data based on a localization approach.

At this layer the information is received from several context sources. It is further organized based on concepts from a predefined model. At his layer the data is organized according to the proposed Context Model. For example, the data from several sensors (GSM, WiFi, Bluetooth) is aggregated into current Location, and the user can experiment with various location algorithms. The user's characteristics are organized based on a FOAF and semantic technologies [16]. We therefore are able to aggregate data into models describing actual relations between users, inferring information about their interests and activities. In an academic environment this allows defining rules specific to users interested in particular research area, or belonging to particular classes.

This layer also provides an abstraction that can be used by all applications to access context information. The domain described by the model acts as a contract between the middleware system and consumer applications. The information and services offered by the contextualization services are consumed by two sorts of applications. Autonomous applications can use the services directly to access context information. They control entirely the reaction to context changes.

In addition, we define a third layer, which uses context Rule actions. Changes in the context may trigger different actions on the smart things according to a predefined rule set. The rules are expressed in an XML-based format and are stored in a remote repository. Things are therefore able to dynamically load and execute locally specific rules, depending on context. An example of such a rule is presented in Fig. 6, which notifies the interested and available user about an event in the field of Distributed Systems. The main rule consists of two standard rules combined by the logical operator AND. The first rule retrieves context information regarding user agenda or university timetable.

```
<rules-config>
      <rule-definitions>
            <rule-def name="DistributedSystemEventNotification"
                  action="category.EVENT_NOTIFICATION">
                  <rule name="userIsFree" />
                  <operator name="AND" />
                  <rule name="userHasInterest" />
            </rule-def>
      </rule-definitions>
      <rule-implementations>
            <rule-impl name="userIsFree" class="rules.StringFieldEquals">
                  <property name="argField" value="CURRENT_ACTIVITY"/>
      <property name="target" value="free"/>
      </rule-impl>
      </rule-implementations>
      <rule-implementations>
            <rule-impl name="userHasInterest"
                  class="rules.StringFieldContainedInList">
                  <property name="argField" value="INTERESTS"/>
                  <property name="target" value="Distributed Systems"/>
            </rule-impl>
      </rule-implementations>
</rules-config>
```

Fig. 6 Example of a context rule

Finally, the fourth layer is responsible with the applications, expressed as rules and actions, which can be used for orientation, information and recommendation purposes. At this layer there are local utilities that can help with context-triggered actions. There are also the applications that use the context data to improve response to stimulus (an interior or exterior request). An application can react to changes in the current context and take specific actions depending on some predefined rules. For this, conditions are evaluated period as the data is retrieved. Third party applications and services can use the API provided by the context-aware services. They can use functions for obtaining particular context data, using filters, or can subscribe for context data. They can also declare new execution rules for users to install on their mobile devices.

2.5 CAPIM Context Model

The CAPIM's context model (Fig. 7) characterizes the situation of an entity. Entity describes any person, place, or object that is considered relevant to the interaction between the user and the environment. In accordance, the context is the collection of information used in some particular form. Thus, the context model includes external data (relative to the environment of the mobile device executing the application, such as location, proximity) or internal information (to the device, such available disk space, battery, capabilities, etc.). The proposed context model aggregates this information into a unique set of data. The context is build based on all detectable and relevant attributes of the mobile wireless device, the application's user, the device's surrounding environment, and the interaction between mobile devices (creating an ad-hoc social interaction map).

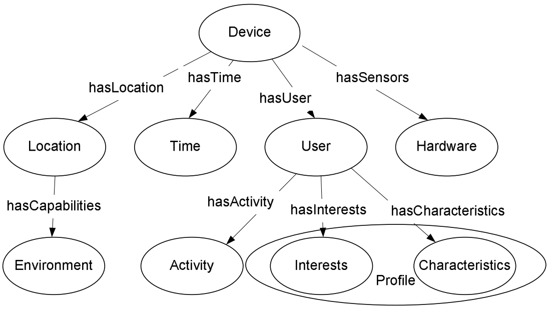


Fig. 7 CAPIM's context model

The context model is hierarchical. On the first layer is the device object (thing) grouping together location, time, identity of a possibly user (in case of smartphones), and the information gathered from various hardware sensors. The device object also provides static information about the device, such as its identifier, operating system and platform, etc.

Location is obtained from several sources. For out-door location we use the GPS or GSM capabilities of the device. For in-door location we combine information received from several sensors, such as GSM cells, Wi-Fi access points, and hardware devices capable of recognizing Bluetooth pairing. The platform also allows experimenting with various in-door locality algorithms and solutions. In this case first the user constructs a module (if one is not already available) for collecting information from sensors. It then aggregates the information into a recognizable form of location data (e.g., the user is in front of a predefined room).

For smartphone, the context information includes information about the user. User's identity is made available from the certificates installed on the mobile smartphone. When the user's identity is found, it is augmented with other information, such as the user's profile and activities. User's activities are discovered from his/her agenda, or from the user's academic schedule (if the user is a student, based on his certificate the schedule is discovered by interrogating the university's data management system). The profile context could include information related to user's research interests, academic interests, or social interests. For the research interests a special service collects and aggregates data from scientific research databases and provides a set of features including automatic collection of information, guided and focused retrieval of researcher profiles, aggregation and storage of structured data in time, aggregated and personalized view of collected information.

The user's profile is provided in either a static form (for example, based on the certificate the user's current academic profile can be easily extracted from the university's digital record database), or is inferred from social networks. For this, the application uses as data sources several social networks: Facebook, LinkedIn. These sources provide dynamic information about user's interests for example. But they also provide information about social relations between users. So, instead of asking users to insert their social preferences again, we learn them from the users' social networks and devise new connections based on the supplementary context information. This allows making queries to the system asking for the whereabouts of the user's current friends, representing users with current interests situated in the immediate proximity, or finding friends that can serve some specific events.

The context also includes system information, collected from specific sensors for battery level, light intensity, accelerometer, etc. The hardware context includes information gathered from external sensors in the environment.

CAPIM's vision is to use the context information as part of the processes in which things are involved. The context can support the development of smart applications capable to adapt based on the data relevant to the location, identity, profile, activities, or environment (light, noise, speed, wireless networking capabilities, etc.). We propose the use of a context model that includes these parameters. Based on this model we propose building smart and social environments capable to adapt to context using mainly the sensing and processing capabilities of smart objects.

CAPIM uses a semantic-based context model, but other models are also supported. For example, data is collected as time series, for long-term and near real-time processing guarantees. The semantic model provides a vocabulary to represent knowledge about context and to describe specific situations. The Context Ontology defines a common vocabulary to manage and share context data. The advantage of such an approach is sharing a common understanding of information to users, devices and services, because the ontology includes machine-interpretable definitions of basic concepts and relations.

The aggregation and semantic services are running on server-side because the semantic aggregation involves collecting and aggregating together data from multiple sources. All things send context information to the aggregation service, where it is further managed and semantically organized. The aggregation service is also responsible to infer the stored data and send aggregated information back to things or applications. The aggregated semantic data is kept in a semantic database. CAPIM's repository implementation uses the Jena Semantic Web Toolkit. The framework provides functions to add, remove, query even to infer data on generic RDF models.

The context ontology captures all context information and models the basic concepts of things, interests and activities, describing the relationships between these entities. For example, for the pervasive computing which can be divided in a collection of sub-domains (e.g. home domain, school domain), we composed our own ontology using domain-specific ontologies. Considering its specific characteristics, CAPIM things' characteristics are organized based on the FOAF ontology (Fig. 8). In this way one can describe things, contexts and activities, and relations to other things. To model a paper or a book, CAPIM uses the PUBL ontology, storing and linking in this way information such as authors, publishing company or the release date. To describe events, dates or locations CAPIM uses the ICAL and GEO / WAIL ontologies.

The main benefit of using domain-specific ontologies is that one can dynamically plug and unplug them from the model when the environment has changed. The CAPIM's ontology is based on other, already implemented ones because in this way the redundancy can be avoided and the semantic stored data can be easier linked with other information on the Web.

Using the context ontology in a CAPIM academic scenario, for example, one can query and infer the stored data finding out new useful information easier. To illustrate the modeling concept we can describe a typical scenario: to socialize, in a break, first year computer science student Tom wants to discuss about Semantic Web with his interested mates. For this he just needs to use CAPIM service. It will interrogate the aggregation service, which will send to device the required data. With a relational model, the service should have to iterate through all CAPIM users, to find their locations and their interests. This semantic model has all this data linked, so the result is obtained faster without being affect its validity.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:org="http://www.w3.org/ns/org#">

      <foaf:Person rdf:about="andreea.starparu">
            <org:memberOf rdf:resource="Gr341C3"/>
            <foaf:name>Andreea Starparu</foaf:name>
            <foaf:nick>andreea.starparu</foaf:nick>
            <foaf:interest>Semantic_Web</foaf:interest>
            <foaf:interest>Distributed Systems</foaf:interest>
            <rdfs:subClassOf rdf:resource="prezentare_licenta"/>
            <wail:location>
                  <geo:Point>
                    <geo:lat>47.235</geo:lat>
                    <geo:long>25.581</geo:long>
      </geo:Point>
            </wail:location>
      </foaf:Person>
      <ical:vevent rdf:about="prezentare_licenta"/>
         <ical:summary>thesis presentation</ical:summary>
         <ical:dtstart rdf:datatype="xsd:data">2011-07-11</ical:dtstart>
         <ical:dtend rdf:datatype="xsd:data">2011-07-15</ical:dtend>
         <ical:location>
               <geo:Point>
                     <geo:lat>47.235</geo:lat>
                     <geo:long>25.581</geo:long>
      </geo:Point>
         </ical:location>
      </ical:vevent>
</rdf:RDF>
```

Fig. 8 Example of FOAF-based description of context in CAPIM

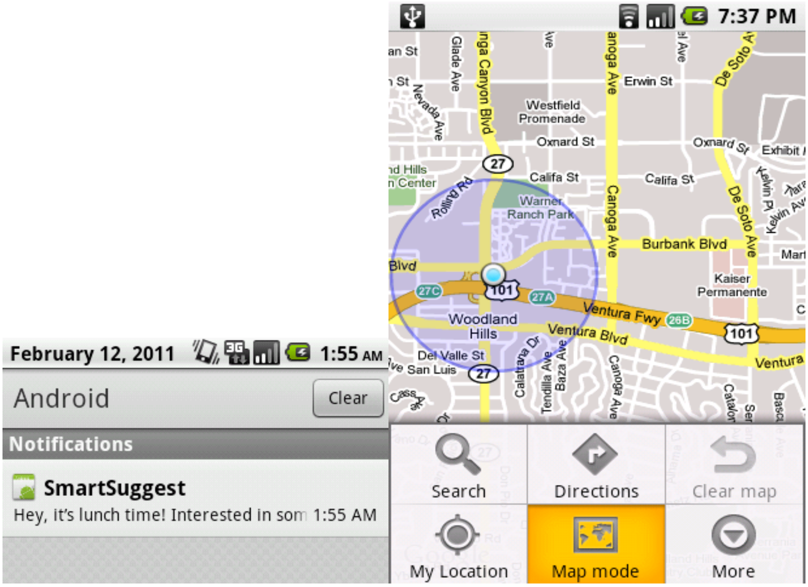
```
<?xml version="1.0" encoding="UTF-8"?>
<rules-config>
      <rule-definitions>
            <rule-def name="showRestaurantSuggestion"
            action="category.PLACE_SUGGESTION"
            parameter="restaurants">
            <rule name="isLunchTime" />
      </rule-def>
</rule-definitions>
<rule-implementations>
      <rule-impl name="isLunchTime" class="rules.IntFieldBetween">
            <property name="argField" value="TIME"/>
            <property name="targetStart"  value="13"/>
            <property name="targetEnd" value="14"/>
      </rule-impl>
</rule-implementations>
</rules-config>
```

Fig. 9 A context-based rule example

2.6 Use Case

A possible application of the proposed platform and context services is an automated support for people in an university, who may be endowed with a portable device which reacts to changes of context by (a) providing different information contents based on the different interests/profiles of the visitor (student or professor, having scientific interests in automatic systems or computer science, etc.), and on the room he/she is currently in; (b) learning, from the previous choices formed by the visitor, what information s/he is going to be interested in the next; (c) providing the visitor with appropriate services – to see the user's university records only if appropriate credentials are provided, to use the university's intranet if the user is enrolled as stuff; (d) deriving location information from sensors which monitor the user environment; (e) provide active features within the various areas of the university, which alerts people with hints and stimuli on what is going on in each particular ambient.

The proposed context-aware platform can be used for the experimental evaluation of many solutions. Users can evaluate methods for gathering context information, for aggregating data using semantics, ontologies. In addition, the platform allows experimenting with complementary context-aware solutions. Consider for example a security component designed to offer a session establishment mechanism, along with session verification processes. Services might use it to verify the identity/authorization of the user currently being the possession of the smartphone. A session can be established through HTTPS using certificate authentication. The solution can, for example, allow users to unlock doors within a building without the requirement of using a physical key or any other replacements (smartcards, swipe cards, etc.). All that is required is a smartphone present in the proximity of the door and a valid user X.509 certificate installed within the phone.



SmartSuggest

Hey, it's lunch time! Interested in som 1:55 AM

Fig. 10 Expanded notification (up), followed by suggestion of nearby restaurants (down)

Another example is based on the rule described in Fig. 9. In this example the context is evaluated. When it is lunch time (anywhere between 13 and 14 hour), the rules triggers an action which, based on the user's current location and using Internet services, finds all restaurants nearby. A notification is then brought up. If the user is interested he can access more details about the suggested nearby restaurants. In another situation, the application observes that the user is in a free time interval according to his/her agenda and place (location), and also that the weather is sunny (using weather Internet services). According to the user's settings it can suggest parks nearby, or other similar outdoor activities close to the user's current location. An example of an execution of the rule is presented in Fig. 10. As a result of the execution, the user is presented with a notification and restaurants suggestions nearby current location.

3 Future Trends and Research Directions in Internet of Things Infrastructure and Services

Internet of Things is not yet a reality, "but rather a prospective vision of a number of technologies that, combined together, could in the coming 5 to 15 years drastically modify the way our societies function" [13]. The evolution of the IoT on medium and long term unleashed a huge interest and gave rise to many research projects, workshops, and conferences, and to the elaboration of reports and survey papers. In this section we discuss the aspects related to the IoT infrastructure and services with emphasis on the main challenges.

It is estimated [32] that IoT will have to accommodate over 50,000 billion objects of very diverse types and technologies. Standardization and interoperability will be mandatory for interfacing them with the Internet. New media access techniques, communication protocols and standards shall be developed to make thing communicate with each other and people. One approach would be the encapsulation of smart wireless identifiable devices and embedded devices in web services. Some initiatives regarding Web services and things' context [33], interacting with the SOA-Based Internet [35], efficient REST-based communications among embedded systems [36] and others demonstrate the high potential of this solution. They also show enhancing the quality of service aspects like response time, resource consumption, throughput, availability, and reliability is possible. The discovery and use of knowledge about services' availability and of publish/subscribe/notify mechanisms would also contribute to improving the management of complex thing structures.

The huge number of things will make their management a very difficult task. One solution is enhanced monitoring facilities to track things and people, and gather information about their status and situation to support informed decisions. Another one is to enable things' adaptation, autonomous behavior, intelligence, robustness, and reliability. They could be based on new general centralized or distributed organizational architectures or on endowing things with self-* capabilities in various forms: self-organization, self-configuration, self-Healing, self-optimization, and self-protection. As an example, in the BIONETS European project [17], evolutionary techniques are embedded in system components to achieve fully autonomic behavior and to properly solve networking and service management issues.

New services shall be available for persistent distributed knowledge storing and sharing, and new computational resources shall be used for complicated tasks execution. Actual forecasts indicate that in 2015 more than 220 Exabytes of data collected from sensors, tracking systems or generated by smart things will need to be stored [32]. At the same time, optimal distribution of tasks between smart objects with high capabilities and the IoT infrastructure shall be found. Since the volumes and rates of these data are very dynamic, elastic Clouds are the best candidates for storing them. Obviously, Clouds can be also used for rapid processing of information and results delivery to the end user.

New mechanisms and protocols will be needed for privacy and security issues at all IoT levels including the infrastructure. Solutions for stronger security could be based on models employing the context-aware capability of things, and on the capabilities of the wireless channels to ensure security.

New methods are required for energy saving and energy efficient self-sustainable systems. Researchers will look for new power efficient platforms and technologies and will explore the ability of smart objects to harvest energy from their surroundings.

4 Conclusions and Remarks

Actual evolution of the Internet of Things towards connecting every thing on the planet in a very complex and large environment gives raise to high demanding requirements, which challenge the actual and future research. The continuously increasing volume of data collected from and exchanged among things will require highly scalable environments able to support the high resulting network traffic, and offer the necessary storage capacity and computing power for data preservation and transformation. Communication protocols are needed to enable not only the high capacity traffic but also maintain the connectivity between things even in case of transient disconnection of wired or wireless links. Also, new solutions should be found for efficiently store, search and fetch the data manipulated in these environments.

The chapter addresses new research and scientific challenges in context-aware environments for IoT. They refer first to the identification, internal organization, provision of context information, intelligence, self-adaptation, and autonomic behavior of individual things. Then, actual research and main challenges related to IoT infrastructure are discussed, with emphasis on services for context awareness, inter-communication, interoperability, inter-cooperation, self-organization, fault tolerance, energy saving, compute and storage services, and management of things collections and structures. Finally, future trends and research directions for the IoT infrastructure are discussed including performance, monitoring, reliability, safety, survivability, self-healing, transparency, availability, privacy, and others.

Acknowledgments. The work has been co-funded by the Sectorial Operational Program Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreement, POSDRU/89/1.5/S/62557.

References

1. Cristea, V., Dobre, C., Costan, A., Pop, F.: Middleware and architectures for space-based and situated computing. Int. J. Space-Based and Situated Computing 1(1), 43–58 (2011), doi:10.1504/IJSSC.2011.039106
2. Dobre, C.: CAPIM: A Platform for Context-Aware Computing. In: 2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (2011)
3. Dey, K., Salber, D., Abowd, G.D.: A Context-Based Infrastructure for Smart Environments, GVU Technical Report; GIT-GVU-99-39 (1999), `http://hdl.handle.net/1853/3406`
4. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2(4), 263–277 (2007), doi:10.1504/IJAHUC.2007.014070
5. CERP-IoT – Cluster of European Research Projects on the Internet of Things. In: Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S. (eds.) Vision and Challenges for Realising the Internet of Things. European Commission - Information Society and Media DG (March 2010)
6. Chen, Helal, S.: A device-centric approach to a safer internet of things. In: Proceedings of the 2011 International Workshop on Networking and Object Memories for the Internet of Things (NoME-IoT 2011), pp. 1–6. ACM, New York (2011), doi:10.1145/2029932.2029934
7. IEEE Computer Society, IEEE Standard 802.15.4, Web page reference (2011), `https://standards.ieee.org/findstds/standard/802.15.4-2011.html` (accessed on March 2012)
8. Chen, L.-J., Sun, T., Liang, N.-C.: An Evaluation Study of Mobility Support in ZigBee Networks. J. Signal Process. Syst. 59(1), 111–122 (2010), doi:10.1007/s11265-008-0271-x
9. Quoc, N.D., Kim, D.-S.: Performance evaluation of priority CSMA-CA mechanism on ISA100.11a wireless network. Comput. Stand. Interfaces 34(1), 117–123 (2012), doi:10.1016/j.csi.2011.06.001
10. Song, J., Han, S., Zhu, X., Mok, A.K., Chen, D., Nixon, M.: A complete wirelessHART network. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys 2008), pp. 381–382. ACM, New York (2008), doi:10.1145/1460412.1460462
11. Cody-Kenny, B., Guerin, D., Ennis, D., Carbajo, R.S., Huggard, M., Mc Goldrick, C.: Performance evaluation of the 6LoWPAN protocol on MICAz and TelosB motes. In: Proc. of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (PM2HW2N 2009), pp. 25–30. ACM, New York (2009), doi:10.1145/1641913.1641917
12. Charalampos, D., Ilias, M., Philippos, T., Dimitris, L., Gregory, Y.: Patient Fall Detection using Support Vector Machines. In: Artificial Intelligence and Innovations 2007: from Theory to Applications. IFIP International Federation for Information Processing, pp. 147–156 (2007)
13. CASAGRAS EU FP7 Project, RFID and the Inclusive Model for the Internet of Things (2012), `http://www.grifs-project.eu/data/File/CASAGRAS%20FinalReport%202.pdf` (retrieved on March 2012)

14. Haller, S.: Internet of Things: An Integrated Part of the Future Internet, Prague (May 13, 2009), `http://services.future-internet.eu/images/1/16/A4_Things_Haller.pdf` (retrieved on March 2012)
15. Kortuem, G., Kawsar, F., Sundramoorthy, V., Fitton, D.: Smart Objects as Building Blocks for the Internet of Things. IEEE Internet Computing 14(1), 44–51 (2010), doi:10.1109/MIC.2009.143
16. Bardram, J.E.: The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 98–115. Springer, Heidelberg (2005)
17. Miorandi, D., Carreras, I., Altman, E., Yamamoto, L., Chlamtac, I.: Bio-Inspired Approaches for Autonomic Pervasive Computing Systems. In: Liò, P., Yoneki, E., Crowcroft, J., Verma, D.C. (eds.) BIOWIRE 2007. LNCS, vol. 5151, pp. 217–228. Springer, Heidelberg (2008)
18. Herrmann, K., et al.: An Emerging Technology for Pervasive Adaptation. In: Proc. of 2nd IEEE Intl. Conf. Self-Adaptive and Self-Organizing Systems Workshop (SASOW 2008), pp. 108–113 (2008)
19. Costa, P.D., Guizzardi, G., Almeida, J.P.A., Pires, L.F., van Sinderen, M.: Situations in conceptual modeling of context. In: EDOCW 2006: Proceedings of the 10th IEEE on International Enterprise Distributed Object Computing Conference Workshops, pp. 6–16. IEEE Computer Society, Hong Kong (2006)
20. Ye, J., Dobson, S., McKeever, S.: Review: Situation identification techniques in pervasive computing: A review. Pervasive and Mobile Computing 8(1), 36–66 (2012)
21. Henricksen, K., Indulska, J.: Developing context-aware pervasive computing applications: models and approach. Pervasive and Mobile Computing 2(1), 37–64 (2006)
22. Loke, S.W.: Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective. Knowledge Engineering Review 19(3), 213–233 (2004)
23. Kalyan, S., Gopalan, V.S.: Hybrid context model based on multilevel situation theory and ontology for contact centers. In: PERCOMW 2005: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 3–7 (2005)
24. Augusto, J.C., Liu, J., McCullagh, P., Wang, H., Yang, J.-B.: Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home. International Journal of Computational Intelligence Systems 1(4), 361–378 (2008)
25. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications 28(1), 1–18 (2005)
26. Huynh, T., Fritz, M., Schiele, B.: Discovery of activity patterns using topic models. In: UbiComp 2008: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 10–19. ACM, New York (2008)
27. Gu, T., Pung, H.K., Zhang, D.Q.: A Bayesian approach for dealing with uncertain contexts. In: Proceedings of Advances in Pervasive Computing, Coexisted with Pervasive 2004, pp. 205–210. Austrian Computer Society (April 2004)
28. van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In: UbiComp 2008: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 1–9. ACM, Seoul (September 2008)
29. Yang, J.-Y., Wang, J.-S., Chen, Y.-P.: Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural classifiers. Pattern Recognition Letter 29(16), 2213–2220 (2008)

30. Kanda, T., Glas, D.F., Shiomi, M., Ishiguro, H., Hagita, N.: Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. In: UbiComp 2008: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 380–389. ACM, Seoul (2008)
31. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is key. Commun. ACM 48(3), 49–53 (2005)
32. INFSO D.4 Networked Enterprise & RFID, INFSO G.2Micro & Nanosystems, Working Group RFID of the ETP EPOSS. Internet of Things in 2020 (2008) Roadmap for the future, Version 1.1 - May 27, 2008. European Commission - Information Society and Media DG (2009)
33. He, J., Zhang, Y., Huang, G., Cao, J.: A Smart Web Service Based on the Context of Things. ACM Transactions on Internet Technology 11(3), Article 13 (January 2012)
34. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A Survey of Context Modelling and Reasoning Techniques. Preprint submitted to Elsevier (March 27, 2008), `http://www.perada.eu/documents/articles-perspectives/survey-context-modeling-reasoning-techniques.pdf` (retrieved April 2012)
35. Guinard, D., Karnouskos, S., Savio, D.: Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. IEEE Transactions on Services Computing 3(3), 223–235 (2010)
36. Castellani, A.P., Gheda, M., Bui, N., Rossi, M., Zorzi, M.: Web Services for the Internet of Things through CoAP and EXI. In: IEEE International Conference on Communications Workshops (ICC), June 5-9, pp. 1–6 (2011)
37. Christensen, H.B.: Using Logic Programming to Detect Activities in Pervasive Healthcare. In: Stuckey, P.J. (ed.) ICLP 2002. LNCS, vol. 2401, pp. 421–436. Springer, Heidelberg (2002)
38. Yau, S.: Hierarchical situation modeling and reasoning for pervasive computing. In: Proc. of IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, SEUS 2006/WCCIA 2006, Tempe, Arizona, US (2006)
39. Thierry, D.: A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory, Classic Works of the Dempster-Shafer Theory of Belief Functions. In: Studies in Fuzziness and Soft Computing, pp. 737–760. Springer, Heidelberg (2008)
40. McKeever, S., Ye, J., Coyle, L., Dobson, S.: Using Dempster-Shafer Theory of Evidence for Situation Inference. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 149–162. Springer, Heidelberg (2009)
41. Sutter, J.D.: Street Bump app detects potholes, tells city officials (2012), `http://whatsnext.blogs.cnn.com/2012/02/16/street-bump-app-detects-potholes-tells-city-officials/` (retrieved March 26, 2012)
42. Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I.S., Mazura, M., Harrison, M., Eisenhauer, M., Doody, P.: Internet of Things Strategic Research Roadmap. In: Internet of Things - Global Technological and Societal Trends, pp. 9–52. River Publishers (2009)

Service Interoperability in the Internet of Things

Jose Delgado

Abstract. The main service interoperability models (SOA and REST) are currently implemented in the Web with text based technologies (XML, JSON, HTTP), conceived for large grained hypermedia documents. Their extension to the Internet of Things context, involving devices with constrained capabilities and unreliable wireless network protocols, implies using a subset of the features of those technologies and adapting the network and message level protocols. This chapter starts by establishing a layered interoperability framework, from the organizational down to the network protocol levels. Then, it assesses the constraints and limitations of current technologies, establishing goals to solve these problems. Finally, a new interoperability technology is presented, based on a distributed programming language (and its execution platform) that combines platform independence and self-description capabilities, which current data description languages exhibit, with behavior description (not just data), elimination of the need of a separate language for schema or interface description, complete separation of data and metadata (optimizing message transactions) and native support for binary data (eliminating the need for encoding or compression).

1 Introduction

Aside older technologies, the main solutions currently available for service and resource interoperability are the SOA (with Web Services) and REST styles, both usually over HTTP. These are the product of an evolutionary line stemming directly from the early days of the Web, where the distinction between a client and a server was clear, stateless browsing and scalability were primary objectives and the hypermedia document was the interaction unit. That was the original Web, or the *Web of Documents* [1]. Today, the world is rather different:

- People evolved from mere browsing and information retrieval to first class players, either by actively participating in business workflows (*Business Web*) or engaging in leisure and social activities (*Social Web*);
- The service paradigm became widespread, in which each resource (electronic or human) can both consume and provide services, interacting in a global *Service Web*;

Jose Delgado
Instituto Superior Technico, Technical University of Lisboa, Portugal
e-mail: jose.delgado@ist.utl.pt

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 51–87.
DOI: 10.1007/978-3-642-34952-2_3

- Platforms are now virtualized, dynamic, elastic and adaptable, cloud-style, with mobility and resource migration playing a role with ever increasing relevance;
- The distributed system landscape has expanded dramatically, both in scale (with a massive number of interconnected nodes) and granularity (nodes are now very small embedded computers), paving the way for the *Internet of Things* (IoT) [2, 3].

The usual meaning of *Web* implies HTTP based systems, whereas *Internet* means IP based systems. In this chapter, we use *Internet* in a more general sense of *network interconnecting networks*, even if these are not IP based, to cater for the low level devices (such as sensors) and the networks that interconnect them.

Figure 1 establishes two main approaches to achieve the IoT:

- **Top-Down**, by extending current HTTP based technologies to a level as low as the devices' capabilities allow (which is known as *Web of Things* [4], or WoT), with gateways that represent the functionality provided by lower level devices, not HTTP or not even TCP/IP enabled. This is the mainstream solution today;
- **Bottom-Up**, by rethinking the interoperability problem in the envisaged scenario of IoT, not just the Web, and deriving a solution that works in the entire IoT, not just the WoT. The goal is to seamlessly integrate distributed platforms, applications and services, covering the ground from higher down to lower level devices and small footprint applications, as well as providing native support for binary data and asynchronous event processing, without the need to use and interconnect complex technologies. This is the solution proposed by this chapter.

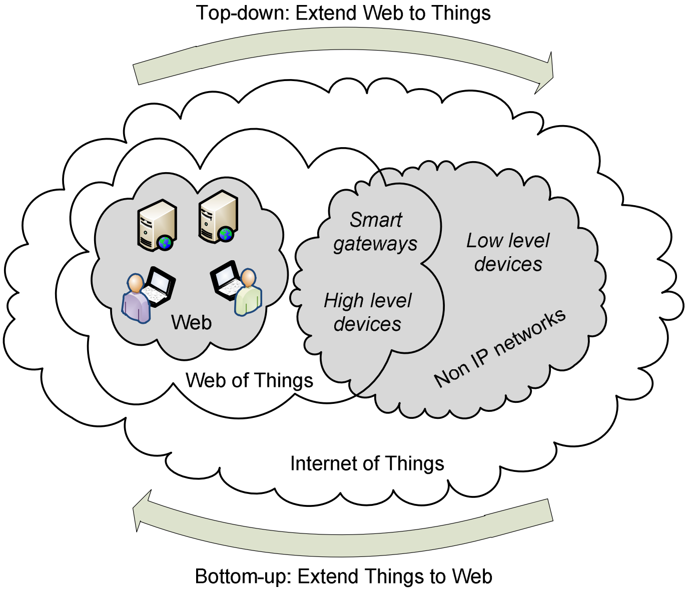


Fig. 1 Approaches to integrate low level devices with people and computers

This chapter is organized as follows:

- First, an interoperability framework is proposed, catering for several levels of interoperability and identifying those dealt with by this chapter;
- Next, the current interoperability technologies at these levels are assessed and their limitations discussed;
- Finally, with the goal of solving the problems previously identified, a language (SIL – Service Implementation Language) and its development and execution environments are proposed.

2 Background

Pervasive computing [5] has been made possible by the ever decreasing cost of microcontrollers. Many of these need to be connected in a network, to generate events that control many applications, as it happens in the case of sensors or readers of RFID tags [6]. Since many applications today are web based, it is only natural to conceive that low level devices can interact directly with conventional web actors, such as people and applications running in servers. All these interconnected entities form what is usually known as the Internet of Things (IoT) [3], as expressed in Figure 1, with the Web of Things (WoT) [4] as the subset that uses HTTP as the underlying message protocol.

The conventional HTTP based solutions for distributed interoperability in the context of Web are the SOA (with Web Services) [7] and REST [8] architectural styles. These are technologies conceived for reliable TCP/IP networks and a reasonable large granularity, with hypermedia documents as the interacting unit and a synchronous, stateless protocol (HTTP).

There is an evident mismatch between this setting and the IoT environment, in which:

- The granularity can be much smaller, with low level devices and very simple messages;
- Communication is more peer to peer (with wireless oriented protocols) than many clients to one server (with omnipresent connectivity);
- The limitations of devices, in terms on power consumption, processing power and memory capacity, have a decisive influence.

Nevertheless, the tendency has been to adapt existing technologies to the IoT context, mainly in the following fronts:

- At the service level in the Web, there is a debate on whether to use SOA or REST [9, 10, 11]. REST is clearly simpler to use as long as applications are not complex, and thus a better match for the very simple APIs found in IoT applications [4, 12, 13, 14]. Nevertheless, there are also SOA applications for the IoT [15, 16]. The DPWS (Devices Profile for Web Services) standard supports a lightweight version of the Web Services stack on resource-constrained devices [17];
- The serialization formats used in the Web (e.g., XML, JSON) are text-based and thus verbose and costly in communications. Technologies have been

developed to compress text-based documents [18], such as EXI (Efficient XML Interchange) and others [19]. However, these are compression technologies, which need text parsing after decompression. Recent native binary serialization formats, such as Protocol Buffers and Thrift [20], do not have this problem;

- There are several efforts targeted at endowing simple devices with IP capability, in particular in the wireless realm and even in IPv6 [21, 22]. The IETF 6lowpan working group has produced a specification (currently a standard proposal) for IPv6 support on Low power Wireless Personal Area Networks (6LoWPAN) [23], such as those based on the IEEE 802.15.4 standard [24].

Simplicity and efficiency (which translates to low processing requirements and promotes scalability, when needed) seem to be the driving forces today, behind not only the IoT and the WoT but even the Web in general as well. This has fueled the generalized adoption of the REST style [9], except for complex, enterprise class of applications, under the assumption that it reduces coupling with respect to the RPC style that is common in SOA based applications [8]. However, interoperability needs to be considered at several levels, as discussed in [25, 26], which is the basis for the interoperability framework described in this chapter and the reasoning that leads to a different conclusion.

Building on the simplicity of REST, the CoAP (Constrained Application Protocol) [27] is a specification of the IETF working group CoRE, which deals with constrained RESTful environments. CoAP includes only a subset of the features of HTTP, but adds asynchronous messages, binary headers and UDP binding.

Compatibility is both a bonus and a curse. All the adaptations towards supporting Web technologies in lower level devices and wireless networks specify subsets of and changes to those technologies. In the end, it is questionable what is gained by this partial compatibility (which ends up demanding changes to middleware and applications) and what is lost by not designing a model that contemplates from scratch (by design) the needs of the modern distributed applications, including the IoT. This is the underlying thought that constitutes the background scenario and motivation for this chapter.

3 An Interoperability Framework

In a distributed world, there is no direct intervention from one resource on another. The only way for a resource to interact with another one is to send it a message, with the requirement that there is a channel interconnecting them. This channel can involve one or more networks, as illustrated by Fig. 1.

In this context, we make the following informal definitions:

- A *resource* is an entity of any nature (material, virtual, conceptual, noun, action, and so on) that embodies a meaningful, complete and discrete concept, making sense by itself while being distinguishable from, and able to interact with, other entities. This means that each resource must have a unique way to identify it (such as a URI);

- A *service* is the interface of a resource, expressed by the set of messages that can accept and react to. A resource implements its service. Resources are the entities that actually exchange messages. However, since a service is the visible face of a resource, saying that a service sends a message to another (or mentioning *service interoperability*) is a common and acceptable language liberty;
- A *transaction*, in this context, is the set of actions and effects resulting from:

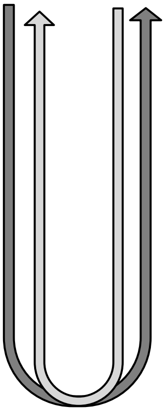
1. Sending a request message;
2. Reaction from the resource that receives it;
3. Sending back a response message;
4. Reaction to that response by the resource that has sent the request.

The resources involved in a transaction perform the roles of *consumer* (sends the request) and *provider* (executes the request and sends the response). In another transaction, the roles can be reversed.

Achieving interoperability between two resources is much more than sending a set of bytes as a message. The interacting resources need to agree on compatible protocols, formats, meanings and purposes, so that a request message sent by one resource produces the intended effects on the other resource and a suitable response message is returned to the former resource.

Table 1 establishes a framework for interoperability by defining a set of layers of conceptual abstraction of interoperability, from very high level (the strategies of both resources need to be aligned so that each message fits a common purpose) down to very low level (the message must physically reach the other resource).

Table 1 Levels of interoperability in a simple transaction. ▬- Request. ▭- Response

Concept	Consumer	Channel	Provider	Interoperability
Alignment	Strategy		Strategy	Organizational (purpose)
Cooperation	Partnership		Partnership	
Outsourcing	Value chain		Value chain	
Ontology	Domain		Domain	Semantic (meaning)
Knowledge	Rules		Rules	
Contract	Choreography		Choreography	
Interface	Service		Service	Syntactic (notation)
Structure	Schema		Schema	
Serialization	Message format		Message format	
Interaction	Message protocol		Message protocol	Connectivity (protocol)
Routing	Gateway		Gateway	
Communication	Network protocol		Network protocol	

A consumer or a provider can be a very complex resource, such as an enterprise, or a very simple one, such as a temperature sensor. Naturally, the higher interoperability levels are more relevant for the more complex resources, but even a mere sensor has a purpose and implements a (very simple) strategy that must fit and be aligned with the strategy of the system that uses it.

We should also bear in mind that the IoT entails not really just interconnecting physical devices but rather fitting them into the overall picture, up to the organizational level. A simple RFID tag can be fundamental in supporting the automation of business processes.

The interoperability levels considered in Table 1 result from a refinement and adaptation to the IoT context of the frameworks described in [25, 26] and can be organized, from top to bottom, in the following categories:

- **Organizational.** Each request from a consumer to a provider must obey a master plan (the strategy). The consumer must know why it is placing that request to that provider, which must be willing to accept and to honor it. Both strategies need to be aligned and implemented by a cooperation supported by complementary activities (outsourcing), as part of a value chain. A key concept at these levels is Enterprise Architecture [28];
- **Semantic.** Both interacting resources must be able to understand the meaning of the data exchanged and the reaction of the other resource to each message sent. This means compatibility in ontology, rules and workflows, so that these resources can participate with matching roles in some choreography;
- **Syntactic.** This category deals mainly with form, rather than content. An interface needs to be defined for each resource, exposing some structure, both at the data and behavior levels, according to some schema. WSDL and REST APIs [29] are examples. The contents of messages need to be serialized to be sent over the channel and the format to do so (such as XML or JSON) also belongs to this category;
- **Connectivity.** The main objective in this category is to transfer the message's content from one resource to the other. This usually involves enclosing that content in another message with control information, implementing a message protocol (such as HTTP), which is then sent over a communications network, according to its own protocol (such as TCP/IP or ZigBee [30]). When different networks (eventually, with different protocols) or network segments are involved or proxies are needed, gateways are used to implement routing and/or protocol adaptation [31].

Each transaction must satisfy interoperability at all levels, both in the request and the response. In Table 1, interoperability at each level (horizontally, in both consumer and producer) abstracts lower levels and ignores higher levels. For example, interoperability at the service level assumes that the correct service is targeted, the correct operation is invoked with adequate parameters and that operation produces the expected result. This ignores the semantics and purpose of the transaction (which must be dealt with at higher levels) and relies on compatible formats and protocols for communication (ensured by lower levels).

In practice, most existing systems specify and implement interoperability in the syntactic and connectivity categories in detail, dealing with the organizational and semantic categories essentially at the documentation level. This is particularly true in the context of the IoT, in which many of the interconnected resources are low level devices, with limited capabilities and intelligence, and consumers and providers can reside in different networks, with different protocols, as shown in Fig. 1. For this reason, and to limit the complexity involved, this chapter deals essentially with the syntactic and connectivity categories, from the service interoperability level downwards.

Fig. 2 illustrates the main steps involved when a consumer in one network performs a transaction with a provider in another network, with a gateway to bridge the two. This could be, for example, the case of a computer application establishing a dialog with a smart phone, reading information from a sensor or controlling some device in a Bluetooth or ZigBee network.

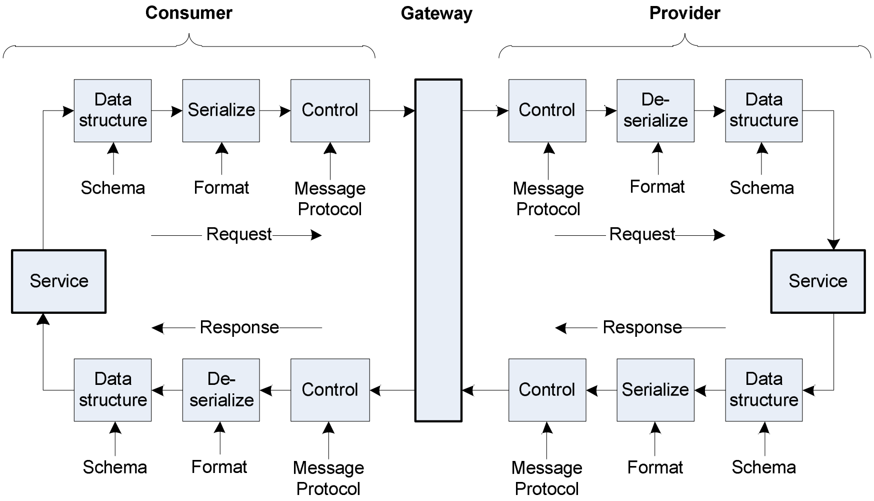


Fig. 2 Main steps involved in a transaction between two services

Sending a request message from the consumer to the provider (top part of Figure 2) entails the following main steps:

- At the consumer:
  - Build the request data structure, according to the request schema;
  - Serialize it, using a format (e.g., XML) that supports that schema;
  - Enclose it as a payload in a suitable message level protocol (e.g., HTTP);
  - Send the complete message;
- At the gateway:
  - Convert protocol information (if needed);
  - Convert the message payload (if needed);
  - Route the message to the provider's network;
- At the provider:
  - Recover the message payload, using the provider's protocol;
  - Rebuild the request data structure, using the provider's schema;
  - Check if the message payload can be accepted;

These steps correspond to the U shape in Table 1 and show how to make services interoperable we need to ensure compatibility at the schema, serialization format and message and network protocols. The trivial solution is to use the same specification at each level (for example, XML reader and writer must use the same schema), but this translates to coupling, an undesirable constraint in distributed systems. This chapter shows how to alleviate this problem.

The third part of the transaction, sending a response to the consumer, is similar to sending the request to the provider, but now the message flows the other way (bottom part of Figure 2). Note that schemas, formats and protocols may not be the same or use different rules with respect to the first part of the transaction (sending the request).

Assuming that the gateway performs its role as a bridge between network protocols, if needed, making the consumer and provider interoperable in syntactic and connectivity terms means solving the interoperability problem at the levels of service and below, as shown in Table 1. These are the levels used in this chapter to discuss interoperability.

4 Assessing Current Interoperability Technologies

We will now use the interoperability framework described above to discuss the suitability and limitations of the main existing solutions to support the IoT. We do not assess the levels below message protocol because we aim at providing a solution that works across heterogeneous networks, with different network protocols.

4.1 Service Interoperability

Naturally, resources are free to use whatever conventions they see fit to ensure interoperability at the service (interface) level. However, application specific solutions should be avoided. In the context of IoT, the two most used models at this level are SOA and REST. Although these are abstract models, in practice only the two most common instantiations, SOA with Web Services and REST with HTTP, are considered. Therefore, the SOA and REST acronyms in this chapter refer to these instantiations.

We will use an example to make the differences between SOA and REST clearer. Figure 3 describes a typical application in the IoT, involving electronic commerce and logistics [6], with a set of processes involved in purchasing and delivering a product. The customer orders and pays (with an option to cancel) a product at the web site of the seller, which sends the product to a distributor via a transport company. The customer can track the evolution of the product during transport, thanks to information from an RFID tag in the product's parcel. When the parcel passes by a RFID sensor (in a truck or distribution station), a signal is sent to the business process at the seller company.

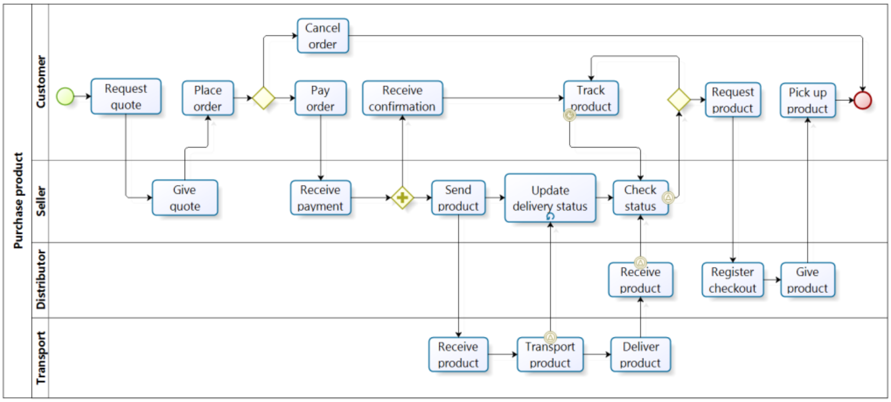


Fig. 3 An example of an application in the IoT

4.1.1 The Essence of SOA

SOA leads to an architectural style that is an evolution of the RPC (Remote Procedure Call) style, by declaring and invoking operations in a standard and language independent way. Compared to RPC, coupling has been reduced and interoperability increased. However, the following principles still apply:

- The resources that model the problem domain are essentially those that correspond to nouns;
- Web Services model each of these resources, exposing a set of operations (its interface) that implement its behavior;
- Only behavior structure (operations) is exposed by the service. State structure (data) is private, implementation dependent and not externally accessible;
- Resources are peers, in the sense that each resource can both offer a service and invoke other resource's service;
- To use a resource, its service description (i.e., WSDL document) must be obtained first, so that a contract can be established between that resource (the provider) and the resource that uses it (the consumer). That contract is static (design time).

Figure 4 partially illustrates the SOA solution to the problem of Figure 3, by describing the application from the viewpoint of the customer (how it progresses and interacts with the seller and the distributor companies). Each of the interacting entities is modeled as a resource, defining a Web Service with operations as needed to support the corresponding functionality. The resulting processes will perform a choreography, as shown in the sequence diagram.

4.1.2 The Essence of REST

A contract between resources, which is an expression of their coupling, is seen as a strong disadvantage by REST proponents, which contend that a more dynamic

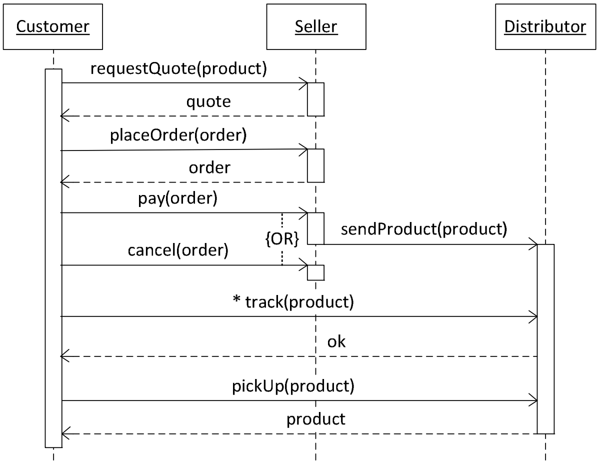


Fig. 4 Interaction of the customer with the seller and distributor, SOA style

approach reduces complexity and coupling, while promoting changeability, scalability and performance [9, 32].

REST is an architectural style that essentially defines a set of best practices of HTTP usage. The most relevant principles, expressed as architectural constraints [33], are the following:

- **Client-server Dichotomy.** There is a clear distinction between client and server roles, with the implicit assumptions that clients are the only ones allowed to take the initiative of sending requests and usually greatly outnumber servers. A response from a server resource is considered a representation of that resource and not a resource in itself;
- **Stateless Interactions.** Each request must include all the information needed for the server to process it. Servers do not store session state, which is only maintained by the client, which means that, in each response to a request, the server must return all the information that may be needed for the next request;
- **Uniform Interface.** This means having the same set of operations (with GET, PUT, POST and DELETE as the most common) for all resources, albeit the behavior of operations can differ from resource to resource. Therefore, at the syntactic level, all resources in REST implement the same service (interface);
- **Explicit Cacheability.** All responses by a server must define whether they can be cached at the client or not. If yes, they can be reused in future equivalent requests, which improves performance and scalability.

The REST style has been inspired by the class of applications that motivated the conception of the Web, in which typically there are many clients accessing a server and scalability is of paramount importance. This justifies the stateless interaction and cacheability constraints, since the load on the server becomes less dependent on the number of clients.

However, the most distinguishing constraint of REST is the uniform interface, which breaks the dichotomy between nouns (objects) and verbs (operations) that is typical of SOA modeling, which stems from object-oriented modeling (usually expressed in UML), in which a specific set of verbs is used to describe a noun concept. In REST, there is only one structuring dimension (resource composition) and any operation that does not fit the semantics of the four basic HTTP verbs is modeled as a resource defined inside another resource.

The rationale for the uniform interface is a logical consequence of the stateless interaction. If the state of the interaction is exchanged between the client and the server in each request-response, what to do next by the client depends on that state and the possible state transition paths, leading to a state machine processing style to model behavior. Each server response is equivalent to a *closure* [34] that includes the necessary information (including links to other resources) to change to the next application state.

This means that a client should not rely on previous knowledge about the server´s resource structure, so that changes in the application can automatically be used by the client without changing it. The client needs only to know how to change state, by pursuing links and using the uniform interface. It is the resource representations retrieved from the server that present the states to which the client can go next.

This is what is usually known as HATEOAS (Hypermedia As The Engine Of Application State) [33, 35], in which the client analyzes the server's response and typically proceeds by sending a request through one of the links contained in it, leading to a state diagram traversal which overall corresponds to a process, as Figure 5 illustrates.

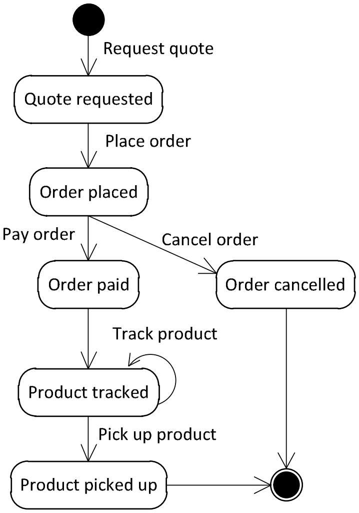


Fig. 5 Interaction of the customer with the seller and distributor, REST style

This is the REST solution to the customer interaction with the other actors of the system in Figure 3, equivalent to the SOA solution of Figure 4. Resources corresponding to actions (such as placing a purchase order) are created at the server and links to them (such as http://company.com/order/product123) are returned in a response to the client. Following these links has the effect of executing the corresponding actions. Such resources are created dynamically, as the state machine progresses. The different actions in Figure 5 are mapped onto a unique set of operations executed on different resources.

4.1.3 Assessing SOA and REST

The main idea behind the uniform interface in REST is to separate the mechanism of traversing a state graph (such as the one in Figure 5) from the individual processing actions of each state. If all states were externally identical, then all state transitions would obey the same rules (HTTP verbs) and the HATEOAS at the client would automatically adapt to changes in the application stored at the server. In other words, the contract binding client and server would not exist or, better said, would be universal and not application specific, with a much smaller coupling than a Web Service contract.

The client-server interaction, however, does not always work in this way and REST cannot ensure this minimal coupling. In practice, the client needs to understand the purpose, meaning and notation (Table 1) of all the responses the server may return, so that it can determine which state to go to next. This means that:

- A generic media type such as XML or JSON is not enough. A concrete schema, with the names used, must be specified and be compatible on both sides, otherwise a representation returned by a resource, for example, will not be able to be analyzed and understood. This is why REST favors standardized media types. However, this hampers variability, which is particularly relevant in the IoT context, given the plethora of interconnected devices. What happens frequently is that a generic media type is used with out-of-band knowledge about the specific schema (implicit information, which can break in case of changes);
- The decision of which link to choose to go to the next state depends on the higher interoperability levels. A person using a browser resorts to additional page information (text, pictures, forms, and so on) to choose a link to click and can even backtrack to follow a different path, if needed. Client applications are less intelligent. In any case, out-of-band information is always needed. The idea that in REST the design-time client-server contract is limited to the data level semantics provided by standardized media types is a mere illusion. This is a direct consequence of dealing with complexity by specifying explicitly only some of the lower interoperability aspects and implicitly assuming the others through names and documentation (informally used to express meaning, subject to misinterpretations);
- REST has a lower resource granularity than SOA, given that internal data and operations in SOA are converted into resources in REST (behavior structure is converted into resource structure). These are simpler and have smaller contracts, but the number of different resource types is higher and the structure of

> links needs to be known to some degree. The overall application contract (client to server coupling) cannot be simpler than the problem itself requires, independently of using a SOA or REST solution.

SOA exhibits a small modeling semantic gap, since interacting entities at the problem level are mapped one-to-one onto resources, but data resource structure is not supported. Interaction contracts are static (design time) and as complex as the functionality of each service requires.

REST has a greater semantic gap, since resources are at a lower level, but supports data resource structure and contracts are simpler and dynamic (although more numerous).

In fact, SOA and REST have dual models, constituting two complementary ways of solving a given problem. Comparing Figures 4 and 5, we can notice that, apart from process specific details, the flow of activities of the client in one figure is dual of the other's. SOA uses activity based flow, with activities in the nodes and state in between, whereas REST makes transitions from state to state, implemented by the activities in between.

SOA is guided by behavior and REST by (representation of) state. In UML terminology, SOA uses a static class diagram as a first approach, to specify which resource types are used and to establish the set of operations provided by each, whereas REST starts with a state diagram, without relevant concern about distinguishing which state belongs to which resource. In the end, they perform the same activities and go through the same states. This should not be a surprise since, after all, the original problem, described by Figure 3, is the same.

There are IoT applications based on SOA [15, 16], but the RESTful approach seems to be more popular, in particular in the lower range (applications with simple devices) [12, 13, 14]. The main reason for this preference is not really caching for performance (data is constantly changing), nor stateless interaction for scalability (interaction tends to be more local and peer to peer than many clients accessing a single device), nor even a preference for state diagrams over processes for modeling (many devices do not even support multistate interaction, only isolated requests). It seems that the preference for REST is based on plain simplicity. Using basic HTTP is much simpler than using Web Services and even link usage is kept to a minimum. In fact, in typical systems such as those described in [4, 12], the only representations returned with links are the list of devices. In other words, there is only application data structure exposed, not behavior. This is essentially a CRUD (Create, Read, Update and Delete) application [35], rudimentary REST at best.

SOA and REST have different advantages, tradeoffs and constraints, summarized in Table 2, which makes them more suited to different classes of applications. REST is a better match for Web style applications (many clients, stateless interactions) that provide a simpler interface. This is why all the major Internet application providers, including cloud computing service providers, now have REST interfaces. SOA can be a better choice for functionally complex, enterprise level distributed applications, in which the semantic gap becomes prominent.

The ideal would be not having to choose between one or the other, by combining the best of both. Support for portable code would also be desirable, to support code migration. Currently, services need to be implemented in general programming languages or BPEL [38]. The desirable features are expressed in the *Wish list (SIL)* column, which serves as the requirements for a language (SIL) that combines the best of SOA and REST, reaping the benefits while avoiding the limitations inherent to each technology. SIL is described below, in section *Rethinking interoperability*.

Table 2 Comparing characteristics at the service level

Characteristic	SOA	REST	Wish list (SIL)
Semantic gap	Low	Higher	Low
User defined interface	Yes	No	Yes
Contract	Static	Dynamic	Both
Changeability	Low	High	High
Design time support	High	Low	High
Granularity	High	Low	Variable
Complexity	High	Low	Low
Best for applications	Complex, event based	Simple, scalable	All
Exposed structure	Behavior	State and behavior	State and behavior
Platform agnostic code	No	No	Yes
Execution paradigm	Workflow	State machine	Both
Links as closures	No	Yes	Yes

4.2 Schema Interoperability

Resources send requests to each other to invoke a given operation, be it with a SOA or REST approach. These requests and their responses usually contain data, which are serialized, sent and reconstructed upon reception of the corresponding message (Figure 2). The sender and receiver need to interpret those data in a compatible way, which means interoperability at the schema level between the corresponding data structures at both ends.

In many cases, the schema information is limited to the indication of a standardized media type, with specific data component names as out-of-band information. This is particularly true in the IoT, in which JSON is a very popular serialization format, as a simpler alternative to XML. JSON Schema is currently just an IETF draft [36], but is raising interest as a simpler alternative to XML Schema.

XML Schema and JSON Schema share many of their goals and characteristics, as expressed by Table 3. The fact is that even JSON Schema can be too much for the IoT, since usually RESTful APIs [29] just specify JSON as the media type, assuming that concrete component names have been agreed between client and server. This provides little support for interface verification when designing services. We need a better way of specifying schema level interoperability, particularly in the context of the IoT. The requirements to do so are expressed by the *Wish list (SIL)* column.

Table 3 Comparing characteristics at the schema level

Characteristic	XML Schema	JSON Schema	Wish list (SIL)
Separate document	Yes	Yes	Yes
Separate specification	Yes	Yes	No
Same syntax as data language	Yes	Yes	Yes
Compatibility	Sharing	Sharing	Compliance
Compatibility cache	No	No	Yes
Schema compiler	No	No	Yes
Validation	Yes	Yes	Yes
Data binding	Yes	Yes	Yes
Complexity	High	Low	Low
Verbosity/size	High	Medium	Low
Reference format	URI string	URI string	Agnostic
Dynamic link construction	Yes	Yes	Yes

There are several ways to improve the schema level interoperability, by solving the main limitations of XML Schema and JSON Schema, such as:

- Do not require both writer and reader of a document to use the same schema, because this requires interoperability for all the documents that satisfy the schema, instead of just the documents that need to be exchanged. Use *structural interoperability* (*compliance* and *conformance*) [37] instead, as a way to reduce coupling and to improve adaptability. The basic idea is to check whether a specific document (not all the documents satisfying a schema) is complies with the schema requirements of the service that will read that document. This is done structurally and recursively, component by component. if it does, the document (a message) can be accepted by the received. This is different from what XML does, which requires that both interlocutors use the same schema;
- A schema document, separate from the data document, is always a huge source of complexity. Although simpler, JSON Schema seems to be following XML Schema's footsteps, which is very revealing of the need for better design time support. However, we can automatically generate a document's schema from its data, which, in consonance with structural interoperability, allow us to get rid of specific schema languages altogether;
- XML Schema and JSON Schema are valid XML and JSON documents, respectively. This means that they suffer from the same verbosity and parsing inefficiencies that text based serialization formats exhibit. It is better to use a schema compiler [20] than a mere compression mechanism [18, 19]. A schema compiler produces binary information that can be used to implement compliance efficiently, both in terms of processing time and size of exchanged information;
- Use a cache-based mechanism to avoid repeating the compatibility checks in every message, if there are no schema changes;
- Do not restrict resource references to URIs, to encompass non TCP/IP based networks, in which nodes are not identified by URIs. A reference should

include a link as a primitive data type, with a format adequate to the network of the target resource. Multi-network references should be supported (including ZigBee addresses, for example).

4.3 Serialization Format

A schema is used to transform the internal data structures into serial messages and vice-versa, as shown in Figure 2. Two of the most common serialization formats are XML and JSON. Both are text based and support schemas. XML retains the look and feel of HTML, with text markup based on start and end tags, whereas JSON delimits data with syntax tokens, with a simple grammar that bears some similarity with data structures in C-like languages. XML has targeted flexibility and generality, whereas JSON has emphasized simplicity, which is after all the secret of its popularity.

In spite of the differences, both suffer from drawbacks and limitations that are particularly relevant in the context of IOT:

- They are text based, which means inefficient parsing and data traversal (all characters of a component need to be parsed to reach the next one), high memory consumption, relevant message transmission times and poor support for binary data. The serialization format should be as close as possible to the data structure, in Figure 2, to minimize the conversion effort in serialization and deserialization. Text has been touted as human readable and therefore advantageous over binary, but this is true only for very simple documents, especially when using XML. Binary compression mechanisms exist [18, 19], but this does not reduce the parsing time, since text is recovered. It would be better to follow the ancient example of programming languages, by using a source format for humans, a binary format for computers and a compiler to synchronize them;
- Metadata is partially interspersed with data, in the form of element/attribute names and tags (in XML). This is redundant with respect to schema information and adds overheads to parsing and transmission times. Data and metadata should be completely separate, much like HTML evolved into XML by separating content from format. This allows to optimize recurring messages, with the same metadata, by just sending the metadata once, caching it with some ID and using that ID in subsequent messages;
- Only data types can be serialized. There is no support for serializing behavior elements (instructions and operations). Describing these in a XML based syntax yields a very cumbersome syntax, such as in the case of BPEL [38]. Incorporating a set of primitive behavior elements and their structuring constructs, at the image of what happens with data elements, is a simple and clear solution of completing message semantics and supporting code migration.

Table 4 summarizes the basic characteristics of XML and JSON as serialization formats, as well as the improvements sought, in the *Wish list (SIL)* column.

Table 4. Comparing characteristics at the serialization format level

Characteristic	XML	JSON	Wish list (SIL)
Data compiler	No	No	Yes
Baseline data	Text	Text	Text, binary
Data delimiters	Markup	Grammar	Grammar, binary tags
Metadata/data separation	Partial	Partial	Complete
Compliance optimization	No	No	Yes
Native binary support	No	No	Yes
Code element types	No	No	Yes

4.4 Message Protocol

In Figure 2, serialized messages need to be sent to the interlocutor, using a message level protocol that adds control information to the message's content.

RESTful applications simply use HTTP, although this protocol has grown to include features all the way up to the service level. It is not a generic message interoperability protocol, since it has been developed specifically for the Web context and with the REST principles (namely, client-server dichotomy, stateless interactions and the uniform interface) as the support for client scalability. This is not the best match for the IoT scenario, which is more based on asynchronous event processing and peer interactions.

HTTP lacks extensibility, headers simply follow a text format instead of a structured format (e.g., XML or JSON), does not support binary data (only text encoded), asynchronous messages, session based interaction for recurring messages, server side initiated requests or client notifications, and follow a strict pattern in message interactions, with a fixed set of operations and of response status codes. HTTP is truly specific for the Web, but with a crucial importance that stems directly from its simplicity, the ubiquity of the Web and the *de facto* generalized friendliness of firewalls.

SOAP, used in conjunction with Web Services, does not incur some of the problems of HTTP but introduces some of its own, namely complexity. Being XML Schema based, is inefficient in every message without benefiting from the flexibility and variability that a schema would allow. SOAP is a standard and does not change frequently. The most common case is to have SOAP over HTTP, but mainly as a transport level protocol, which is another source of inefficiencies.

The message protocol is another level in which substantial improvements can be made. This is expressed in the *Wish list (SIL)* column of Table 5, which also summarizes the main characteristics of HTTP and SOAP.

Table 5 Characteristics of message level protocols

Characteristic	HTTP	SOAP	Wish list (SIL)
Baseline format	Text	Text	Binary
Message is	Character stream	Character stream	Byte stream
Structured headers	No	Yes	Yes
Layered headers	No	Yes	Yes
Schema based	No	Yes	No
Generic operation call	No	Yes	Yes
Generic response status	No	Yes	Yes
Message transaction	Synchronous	Synchronous	Asynchronous
Reaction messages	No	No	Yes
Heterogeneous network support	No	No	Yes

We would like to ally the simplicity of HTTP with the capabilities of SOAP and to throw in some additional features, such as:

- Native support for binary data. This implies using a byte stream as the rock bottom transport format, not a character stream (text). A bit stream is also possible and more compact [19, 42], but requires more processing effort to encode and decode, which is relevant in small IoT devices with low processing power;
- Native support for asynchronous messages and responses, with additional information at the MEP (Message Exchange Pattern) level;
- Reaction messages, sent to a resource without specifying any operation and letting that resource react as it sees fit, by automatically selecting the appropriate operation, based on the type of the message;
- Support for messages spanning heterogeneous networks, with different protocols such as TCP/IP, ZigBee or Bluetooth, without requiring message format conversions in gateways. Essentially, this involves not being dependent on the protocol, even at the level of identification of the resources that messages are addressed to. A link used to identify a resource identification may be no longer a single URI, but rather an inter-network path (a list of URIs or even of other identifiers, such as a ZigBee address).

5 Rethinking Interoperability

5.1 An IoT Case Study Scenario

We envisage a typical scenario of the IoT, depicted in Figure 6a, in which a client application accesses sensors coordinated by a controller and interconnected by a sensor network. In a logical view, the sensors are components of the controller, as shown in Figure 6b, assuming that could be other controllers, coordinating their own set of sensors.

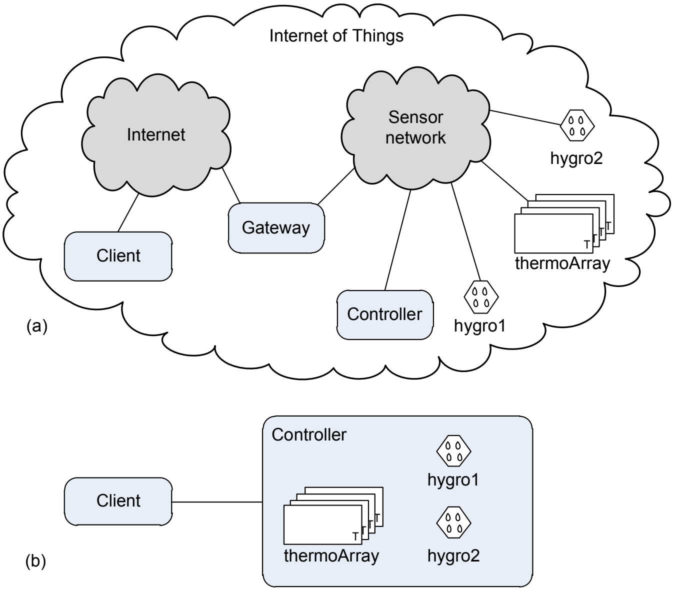


Fig. 6 A client accessing devices in a sensor network. (a) Network view. (b) Logical view.

The usual solutions to implement this scenario would be:

1. To provide an API (REST or SOA) in the Controller, which makes a mashup of the functionality of the sensors [39], hiding them from the client;
2. To endow sensors with TCP/IP and HTTP capability and to implement a REST or SOA service in them [40], so that they can be used directly by the client.

Both solutions have drawbacks:

1. The controller needs to reproduce the functionality of sensors and needs to be changed whenever unforeseen changes are made at the sensors or their configuration;
2. The sensors need to support HTTP and REST or SOA, which requires more capabilities in each sensor, both to support message processing and to overcome the natural mismatch between TCP/IP and the wireless protocol.

Solution 2 seems to be the most popular given that sensors are becoming equipped with better hardware and REST and SOA are tried technologies with implementations at this low level that are starting to appear [12, 16]. However, the fact is that these technologies are not native solutions for IoT (were conceived for text based hypermedia rather than this level of granularity, usually binary based) and they rely on the simplicity of the application.

Another way to put the issue is to say that applications cannot get more complex than the technology allows. History shows that, when the technology evolves and gets better and more efficient, new applications immediately exploit those improvements. That is our motivation.

We will use the *Wish list (SIL)* column in Tables 2 through 5 as a set of requirements to design SIL (Service Implementation Language), a new interoperability technology that considers the IoT range of applications right from the start. The basic tenet is to reconsider the interoperability problem and to derive the solution that most closely matches the context of IoT, without the constraint of compatibility with previous technologies. We continue to use the interoperability framework described in Table 1.

5.2 Service Interoperability

The basic problem is to make two resources able to interact by making their services compatible at the syntactic interface level (assuming higher interoperability levels have been dealt with). According to Table 2, we want to be able to combine the design time support and low semantic gap of SOA (resources with a fixed set of generic operations) with the linked and dynamic resource structure of REST, which includes operations as resources.

To make this possible, we have defined a language to specify and implement resources and their services (SIL – Service Implementation Language), which is able to describe not only data but also behavior (operations). In fact, it is a distributed programming language, supporting both the RPC and resource interaction styles. Not only can each resource implement any number of operations, each with any type of input and output, but it can also be structured, recursively composed of other resources, with a path to identify component resources. Operations are (behavior) resources and can be sent messages.

SIL allows the specification of both platform independent code (SIL instructions) and platform dependent code, with primitive operations implemented in a programming language such as Java or C#. In each distributed node, a SIL server is needed, to host resources and to provide an execution platform, to run SIL code and to interface other programming languages through adapters. This interface can be done statically, with compile time generation of a class or resource description (in a similar way to binding WSDL and object-oriented programming languages), or dynamically, by using reflection.

SIL nodes can use any protocol that provides transport for binary messages. A SIL server can support several protocols and heterogeneous networks if network gateways are available.

Program 1 illustrates some of the features of SIL with a simple description and partial implementation of the *Controller* resource of Figure 6. Like JSON, structured resources are defined between curly brackets and named components with a colon. SIL uses the same mechanism to define named operations (with the keyword **`operation`**), which can be primitive (implemented in another language, with the keyword **`primitive`**). Each operation can have only one input and output parameter (but which can be structured), separated by the token `->`. These can be accessed in the body of the operation with the predefined identifiers **`in`** and **`out`**, respectively.

```
{
// definitions
   define humidityValue as [0..100];   // integer range
   define thermometer as {
      networkID: integer;     // ID in the sensor network
      primitive (-> number);  // unnamed operation (get temperature)
      setAlarm: primitive ({&any; number}); // structured parameter
   };

// state components
   hygro1: {
      networkID: integer;           // ID in the sensor network
      primitive (-> humidityValue); // unnamed (get humidity)
      setPeriod: primitive ([1..60]);// sets history sampling period
      history: primitive(-> array humidityValue);// get values stored
   };
   hygro2: hygro1;                  // replicate resource
   thermoArray: array thermometer;  // array can grow

// operation components
   init: operation (array integer){ // array has IDs of thermometers
      for i (0..in.size)
         addThermometer <-- in[i];
   };

   addThermometer: operation (integer) {
      th: thermometer;
      th.networkID = in;
      thermoArray.add <-- th;    // add an array element
   };

   getAverageTemperature: operation (-> number) {
      total: number;    // initialized to 0.0
      if (thermoArray.size == 0)
         reply 0;                   // default value
      for i (0..thermoArray.size)
         total += thermoArray[i];
      reply total/thermoArray.size; // compute average
   };
};
```

Prog. 1 Specification of the *Controller* resource in SIL. Reserved words are in bold.

Definitions, at the beginning, are only auxiliary and generate no components until used. Components can also be declared inline, such as shown by `hygro1`.

Operations are behavior resources and are executed (invoked) by sending them a message, with the `<--` operator. The message to send can be omitted, if there is no input parameter. This is the same mechanism used to send messages to non operation resources, such as `hygro1`. In this case, an operation without name with matching input parameter will be invoked. This usually corresponds to a GET operation in REST.

The *Controller* has three operations (`init`, `addThermeometer` and `getAverageTemperature`), all non-primitive. This means that they will be executed by the SIL execution platform, in a portable way. The compiler transforms their instructions into *silcodes* (equivalent to Java's bytecodes), which are executed by a virtual machine (an interpreter). Resources that have no primitive operations can be suspended, migrated from one server to another and have executed resumed at the new server.

Once created and registered in a server's directory (which includes a resource name server), the *Controller* can be configured with a set of sensors by sending its `init` operation a message with the `networkIDs` of several sensors:

```
controller.init <-- {107; 129; 114};
```

This resource path, ending in an operation, is typical of the REST style, but in SIL it blends seamlessly with the SOA style, since we are not limited to an universal set of operations.

The *Client* resource could have a definition such as shown in Program 2. Note that instructions can be interspersed with state resource declarations. The entire set will be executed once, upon resource creation, becoming ready to receive messages afterwards. In effect, the declaration of a resource is its constructor.

```
{
   t: number;
   h: integer;
   t = controller.thermoArray[1] <||;  // asynchronous message
   h = controller.hygro1 <--;          // synchronous message
   controller.thermoArray[2].setAlarm({&alarm; 30});
   someOtherResource <-- {temperature: t; humidity: h};

   alarm: operation (integer){ // networkID (no name for flexibility)
      ...   // deal with alarm
   };
};
```

Prog. 2 Specification of the *Client* resource in SIL

An asynchronous message (with the `<||` operator) invokes the unnamed operation of `thermometer` 1 in the *Controller*'s array but returns a *future* immediately (stored in `t`), so that processing can proceed concurrently with the message request. When the reply value arrives, it will automatically replace the future. If the value of `t` is used before that (in the message to `someOtherResource`), execution of the *Client* will be blocked until the value is available. The message to sensor `hygro1` is synchronous and waits until the value is replied.

The Client has an operation `alarm`, to be invoked whenever a given sensor exceeds some temperature. The operation `setAlarm` in `thermometer` 2 is sent a message composed of a reference to the *Client*'s `alarm` operation (obtained with the `&` operator) and a threshold temperature. The first component of the `setAlarm` operation's parameter is declared also as a reference (again, using the `&` operator), with the predefined value `any`. All resources comply with `any`, which means that any resource can be used to receive that alarm.

SIL is a distributed programming language, in which references are global (such as URIs), not local (pointers). For this reason, references must be specified explicitly with the & operator, as in this example, and, unlike typical programming languages, assignments have structural copy semantics. This means that, in an assignment to a structured resource, only the components of the value to assign that *comply* with the target resource are actually assigned [41]. This is similar to what an XML processor does, by processing only the tags it recognizes and ignoring the rest.

In these examples, resources are declared and referenced with static pathnames, which allows design time support and verification from the SIL compiler. However, it is also possible to create and delete resources dynamically and specify resource pathnames as structured references, with an array of links, which is the basic support for heterogeneous networks, with different protocols and resource identification mechanisms (not shown here for simplicity).

5.3 Schema Interoperability

There are no type declarations in SIL, only values with an associated variability. For instance, `integer` in the declaration `networkID` in `hygro1` (Program 1) is not a type but the predefined value zero with an associated variability identical to the allowed integer range. The `networkID` component gets both the value and the variability. The value can change, but not the variability. In this way, any valid integer value can be assigned to `networkID`. On the other hand, `humidityValue` is defined with a smaller range of variability (ranges verified by the compiler in assignments).

The hygrometer `hygro1` is defined directly, without a previous definition, and `hygro2`, another identical sensor, is simply defined by replication. There is no type instantiation. Actually, type compatibility in SIL is done not by a shared type declaration but by *structural interoperability* (*compliance* and *conformance*) [37], in which only the used components are required. For example, if we need to invoke an operation X, any resource that implements it (with compatible input and output parameters) can be used, independently of all the other components it may have. The reason for this is the distribution context, in which the lifecycles of resources are independent and common named type declarations become meaningless.

SIL resources and messages are completely self-describing, with a schema mechanism that differs from that of usual schema languages in two fundamental ways:

- The schema is specific of a given resource (document, message or service implementation), not of a set of documents. Instead of ensuring compatibility between resources by sharing a common schema, structural interoperability is used to check compatibility whenever needed. This reduces coupling and widens compatibility [37], since now interoperability is checked on a message by message basis, not on a full set of possible messages;
- There is no separate schema language. SIL itself, the declaration of resources themselves, fills this role. This is possible because the schema pertains to one resource only and there are no types, only values with an associated variability.

The SPID (SIL Public Interface Descriptor) is the equivalent of WSDL or WADL and simply consists of all the public features of a resource declaration such as those of Programs 1 and 2. It suffices to remove the body of non-primitive operations and any resources declared as `private` (feature not illustrated here, for simplicity), which the compiler does easily. Design time compatibility checking is ensured by structural interoperability, which the compiler supports.

5.4 Serialization Format

SIL has a source text format, adequate for people and illustrated by Programs 1 and 2, and a binary format, used for computer processing, either to be stored in a file or sent in a message. For programming, the source text is the input and the binary format is derived by the compiler. For runtime generated messages, only binary is used, although a decompiler can generate a source text format. The binary format usually includes metadata that supports self-description.

SIL uses the concept of *binary stream*, a sequence of bits or bytes with content's meaning and format known to both sender and receiver, encoded in a modified version of the TLV (Tag, Length and Value) scheme used by ASN.1 [42]. Streams can be composed of other streams, which allows for instance XML or JSON strings (a string is also a binary stream, with some encoding) to be part of a SIL resource. Each serialized SIL resource can have up to three streams:

- A source stream (a string such as Program 1 or 2);
- A compiled stream, composed of streams corresponding to primitive resources and streams corresponding to structured resources, composed of other streams. No metadata, aside from the semantics resulting from the streams' tags and structuring, is included here;
- A metadata stream, which includes information on component indices (relative position in the resource), names and value variability.

A resource can use the following combination of streams:

- Source only: mostly for programming and documentation (including SPIDs, the equivalent to WSDL and WADL), but can also be sent at runtime and compiled on the fly by the receiver;
- Compiled + metadata: complete information as well (aside source comments), but more efficient than source only;
- Compiled only: the most efficient, but does not support runtime interoperability checks, unless an optimization mechanism is devised such as the one described below;
- Metadata only: used essentially to represent SPIDs more efficiently, in binary format;
- All three: all the information available on a resource.

Using only the compiled stream looks like a return to the old specific binary formats, but it is in fact one of the distinguishing features of the support of SIL to the IoT applications. In this context, messages from or to low level devices typically follow the same schema until the device suffers some change. Sending metadata in each message is rather inefficient. The compiled stream can include a token returned by the server on the first message request. On reception of subsequent requests, the server checks this token and skips interoperability checking if the token matches its own or returns an error (the client then repeats the request with compiled + metadata). This is not a security feature, but a mere optimization mechanism. Its robustness depends on the number of bits of the token and its pseudorandom non-repeating evolution algorithm. Each resource maintains a cache of the tokens returned by the resources it has sent messages to. This mechanism bears similarities with the Etag header of HTTP.

Separating data from metadata is also done in the binary formats Protocol Buffers and Thrift [20], precisely for efficiency reasons. Schemas can be much larger than actual data, a problem identified but not solved in [19]. The solution in SIL is to be able to send a message without metadata, with a mechanism that still ensures interoperability with the help of design-time checks by the compiler.

5.5 Message Protocol

A message in SIL is a resource, just like any other, and can include operations. The message protocol is the simplest possible to allow a message request and reply, independently of message content. All the rest is extensible and uses the envelope approach, in which a message is encapsulated with further control information into another (the envelope) at the sender and retrieved from that envelope at the receiver. Security, for instance, can use this mechanism. There are no specific purpose headers.

The message protocol supports asynchronous messages (with futures that can be cancelled if the client gives up waiting or decides otherwise) and application faults (exceptions). Each SIL server maintains a message ID generator, based on a non-repeatable, large sequence pseudorandom number generator. The message ID is used to correlate a response to the original message sender (which may be blocked, waiting for that response, if the message was asynchronous).

This ID mechanism is similar to the Token option in COAP (Constrained Application Protocol) [27], but more efficient since it is part of the basic protocol and handled in binary. In fact, the SIL message protocol contemplates the most relevant features of CoAP, namely asynchronous transactions and binary control information (headers). In fact, these features should be available in HTTP itself.

The main message types of the SIL message protocol are described in Table 6.

Table 6 Main message types of the SIL message protocol

Message category/type	Description
Request	*Initial request in each transaction*
React	React to message, no answer expected
React & respond	React to message and answer/notify
Assimilate	Merge the message into the receiver resource, subject to structural interoperability (only compatible components are replaced). It bears similarities with PUT in REST.
Ammendment	*Further information on an already sent request*
Cancel	Cancel the execution of the request
Response	*Response to the request*
Answer	A (structured) value returned by the `reply` instruction
Resource fault	A (structured) value returned as the result of an exception
Protocol fault	A status code resulting from a predefined protocol error
Notification	*Information of completion status*
Done	Request completed but has no value to reply
Cancelled	Request has been cancelled

The message protocol includes the addresses of both sender and receiver, so that the request can be addressed and the receiver can address the response back. These addresses, which correspond to resource names and pathnames in SIL, build on the following assumptions:

- Pathnames can span several networks, with different addressing schemes (e.g., TCP/IP and ZigBee);
- A name server exists for each network, so that a pathname can be converted into a list of network addresses, one for each network;
- Gateways inspect the address list in each message and route it accordingly.

The SIL message protocol includes either an address or a list of addresses for both sender and receiver. These are encoding using the same TLV format of the serialization protocol. This means that SIL resources can directly address others in different networks.

6 Assessing the New Approach

6.1 Contrasting SIL and Related Technologies

History has taken its course and evolved from two main technologies, HTTP and HTML, in a context of many clients for each server and essentially retrieval of multimedia documents. Together with the corresponding execution platforms, the server and the browser, they constituted a well matched technology set that fostered the exponential development of the Web.

The mismatches begun when these technologies started being used and extended for all sorts of application domains and scenarios, from large business systems down to small IoT applications, from large scale server accesses to peer level interaction, from synchronous to asynchronous transactions, and so on. Table 7 tries to shed some light into the picture, by expressing the depth span in the interoperability ladder of some of the most relevant technologies.

Table 7 Levels of interoperability tackled by some existing technologies. Lighter gray in the right column means future work.

Concept	Interoperability level	HTTP	XML JSON	SOAP	WSDL	BPEL	REST	SIL
Alignment	Strategy							
Cooperation	Partnership							
Outsourcing	Value chain							
Ontology	Domain							▒
Knowledge	Rules							▒
Contract	Choreography					■		■
Interface	Service	■			■	■	■	■
Structure	Schema	■	■	■	■	■	■	■
Serialization	Message format	■	■	■	■	■	■	■
Interaction	Message protocol	■		■	■		■	■
Routing	Gateway	■		■			■	■
Communication	Network protocol							

HTTP has been enriched with all the features needed to support interaction in the original Web context. It is actually a service level protocol, with a fixed set of operations (such as GET and POST). It does almost everything, including control data description and type of payload data (with Internet Media Types).

XML has generalized HTML, separating data from formatting and introducing self-description with a schema, but retained much of its look and feel, still with a data document nature (just data, instead of a more complete service nature, by including code) and text with markup (lacks native binary support).

Web Services appeared to fill in the service gap, removing the HTTP's restriction of a fixed interface. But, as Table 7 shows, there is a great overlap in interoperability levels between HTTP, SOAP and WSDL. The latter can be bound to protocols other than SOAP and SOAP can be bound to protocols other than HTTP. In practice, the most common situation by far is to have WSDL with SOAP over HTTP, which means that all this generality is seldom used but has permanent costs. SOAP treats HTTP essentially at transport level, ignoring many of its features. This is a sign of mismatch in implementing generic technologies. One sits, as is, on top of another, without separating the components that match from those that don't. This increases complexity and decreases performance.

Another sign of over generality is the universal use as XML as the underlying serialization format, with specific schemas, such as it happens in SOAP, WSDL and BPEL. This is a good thing, in principle, but these schemas evolve rarely and are standardized, which means users cannot change them. Therefore, all the power and flexibility of XML becomes of limited usefulness, at the same time that its verbosity and complexity are always present. A classic example is the assignment instruction in BPEL, with a baroque syntax instead of a mere equal sign.

These problems are particularly stringent in lower level applications, such as those in the IoT context, and justify the increasing popularity of REST and JSON. REST is essentially HTTP, as Table 7 expresses, or a set of best practices on how to use it. JSON is much simpler than XML and a good match to REST. Code still needs to be provided by another technology, typically a generic programming language. This is fine for simple applications but is limited for more complex ones.

In summary, the main problems are:

- The basic technologies, HTTP and XML (or JSON) are already immediately below the application level (the reason why REST is so simple to use). This causes mismatches, duplications and inefficiencies when subsequent technologies, such as Web Services, have to map onto them;
- The use of a single serialization language, for both people and computers, means that it becomes hard to read, inefficient to process and awkward to support binary data.

The purpose of SIL is to cleanup this scenario, learning from previous technologies and showing that a single language can replace many of the existing technologies. The simple description and examples used in this chapter are not enough to fully show how this can be accomplished, but the most fundamental ideas are:

- Move from client-server to peer dialog, in a service oriented paradigm;
- Base interoperability in compliance and conformance, not schema sharing;
- Derive the schema directly and automatically from the document, instead of having a separate document with different rules;
- Support and describe both data and code, as well as service and resource architectural styles;
- Use one serialization format (text based) for people and another (binary based) for computers, but derive the second from the first automatically;
- Use a simple protocol as the underlying communication mechanism.

SIL may be seen as a return to the RPC (Remote Procedure Call) programming model. Up to a certain point this is true, but with fundamental differences:

- Message marshalling (data serialization) has its own rules, which include self-description. This means that different languages can interact. Actually, coupling is even lower than in XML based systems, since compliance and conformance are used instead of schema sharing [37];
- The target of a message is identified by a distributed reference (such as a URI), in a server based interaction setting;

- Asynchronous transactions are readily supported;
- The interaction is not merely data based. Full resources (data + code) can be exchanged, which means that the basic mechanism is not the operation call but in fact the migration of a resource (the message).

To the best of our knowledge, there is no other proposal with such a wide range of objectives, while maintaining the decoupling and distributed interoperability that characterize current Web applications and technologies. Web Services or HTTP+XML/JSON are usually the rock bottom of existing proposals in Internet based contexts. There are, however, some attempts to change parts of the global scenario.

Web Sockets [44] are fundamental in the efficient support for binary data. They use the protocol upgrade feature of HTTP and provide a substantial degree of compatibility with existing systems. Part of the HTML5 world, servers and firewalls are increasingly supporting them.

The textual nature of markup languages has been recognized as inefficient. Using text as the serialization format requires textual parsing at the message receiver and a heavier effort to generate the corresponding memory data structures than when using a binary format [47]. Proposals such as EXI (Efficient XML Interchange) involve compression and decompression for transmission or storage purposes [18], but text must be recovered and therefore parsing time is not reduced. Native binary serialization formats, such as Protocol Buffers and Thrift [20], aim to solve this issue but deal only with data. Binary is also the path taken by SIL, but with support for code. Since SIL has two serialization formats (text and binary), text is better (for flexibility) when a priori knowledge of the interlocutors is low, and binary is better (for performance) when messages repetitively use the same schema and the compiler can be used. Changing between the two formats can be automated by using a token, as described in the two previous sections.

Distributed applications can be programmed in generic programming languages, using XML or JSON for data level interoperability, or higher level languages, such as BPEL. This is most common in complex enterprise applications, typically SOA and XML based, but has been shown to also fit with the REST style [38]. BPEL provides the support for behavior (code) that Web Services lack, but constitute a separate technology with a different paradigm (processes) and a cumbersome XML based syntax that becomes usable only by resorting to visual programming tools.

SIL has a classical syntax and supports not only processes but also services and resources, in an integrated way. Other proposals also favor the resource based approach. In [48], an information centric process model is proposed, centering the resource concept on business entities instead of instances of workflow activity. Others propose to represent and transfer not only data but behavior as well, such as a method to expose process fragments (described declaratively as reusable workflow patterns) as resources and to map business process concepts onto the usual HTTP-style of CRUD operations [49]. Going a step further, in Computational REST (CREST) [50] the basic entities are computational resources, in the form of continuations [51], providing a base model for code mobility. The client is no longer a mere interface to the user but a computational engine, capable of executing these resources. State transfer is a side effect of this execution.

This is precisely what SIL offers, showing that it is possible to combine all these features under a single model and syntax. Tables 2 through 5 compare SIL and other technologies in further detail.

Table 7 also expresses that SIL will be extended to encompass the two topmost semantic levels, initially by using SIL instead of XML to serialize RDF and OWL documents and afterwards by incorporating constructs to express knowledge and ontologies directly in SIL. However, this is future work and outside the scope of this chapter.

The organizational levels, above semantics (Table 1), are better expressed by frameworks and development methods [28] than by languages with descriptive or execution semantics.

6.2 Implementation

We have developed a compiler based on ANTLR [43], which converts source to instructions and data in a binary format, according to the streams described above. An interpreter then executes the binary code (*silcodes*, equivalent to Java's bytecodes). The current implementation, in pure Java, is not optimized and has a performance roughly 50 times slower than a Java Virtual Machine (JVM). However, much of that time is spent just on method dispatch, the mechanism used to execute the various silcodes. A C based interpreter, for example, would be much faster, although harder to develop. To maintain flexibility and control of implementation, we did not use a JVM and bytecodes.

Support for distribution is implemented with a Jetty application server, but any other server will do. In fact, we only need a protocol handler, which can be much simpler in the case of simpler network protocols. For message exchange, we require only a transport level protocol. We currently use Web Sockets [44], with a cache for automatic connection management, but again any lower level message transport protocol will be enough, provided that it implements the level of reliability required by the application or the application provides that itself.

The Jetty server connects to a SIL server (to handle the SIL message level protocol) that hosts a resource directory for service discovery. This is a regular service, just like any other, that contains references to the SPIDs of the resources registered in it. This directory can be searched for a suitable service by supplying keywords and/or a SPID as required by the client. The directory then searches for these keywords in the registered SPIDs and performs a structural interoperability check to ensure that the returned references to SPIDs are conformant to the SPID used in the search. Similarity ranking [45] is not supported at the moment.

Figure 2 shows the basic message path in a transaction between two services, each implemented by a resource. Figure 7 shows the basic architecture of a SIL node, capable of hosting SIL resources. This is the unit of distribution in the SIL realm. A reference to a SIL resource is made of two parts:

- The network level address of the SIL node. This is network dependent. With TCP/IP, this is the IP address of the SIL node;
- The path, within that node, from the directory (the root of the resource tree) to that resource. This is network agnostic and depends only on the structure of the resources.

URIs join the two parts in a single string. SIL has a primitive reference type that maintains the two parts separate, so that these references can be used in non IP networks.

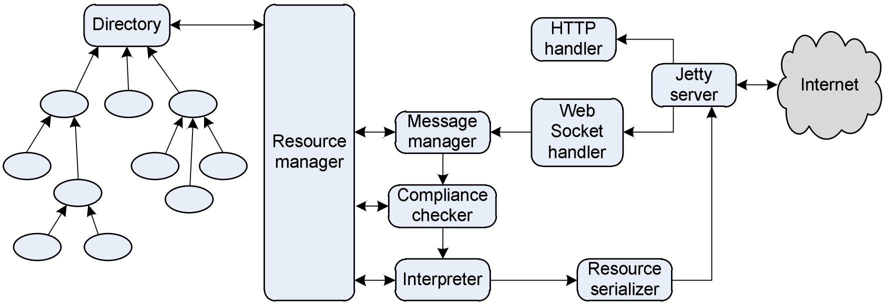


Fig. 7 Basic architecture of a SIL node

The architecture of a SIL node can be described as follows:

- The application server (Jetty, in this case) is the interface to the network. It supports several message handlers, which means that it can deal with several message level protocols. We have only implemented two, HTPP and the message protocol of SIL. Only the steps ensuing the latter are described here. There is a list of handlers to be invoked and each checks whether it recognizes the message format. The first one to do so gets the message for further processing. All SIL messages begin with “SIL”, encoded in UTF-8;
- A message received is handled next by the Message Manager, which determines the recipient of the message, the type of the message (Table 6), which streams are present in the message and, for some message types, whether a compliance token is present (described in section Serialization Format, above);
- The Resource Manager implements the access to the structured resources registered in the Directory, obtaining internal references (indices in a resource table) to resources targeted by messages or by distributed references (URIs, for example);
- The Directory is the root resource and implements several operations, such as searching for a resource which conforms to a given SPID. The resource tree depicted in Figure 7 shows only containment relationships. Any resource can have a distributed reference to another, but only if it is registered as globally accessible in a Directory. This means that resources can be locally reached from others during execution of a SIL program, but only registered resources, directly in the Directory, can be addressed by a global, distributed reference;

- The Compliance Checker performs type compliance between the message and the addressed operation or resource. It can do so in text or binary formats, since each has all the information needed, as long as the metadata stream is also present. Naturally, this is faster when done in binary. Messages that include a compliance token, obtained in a previous checking, can skip this step, as mentioned before in the section Serialization Format;
- If the resource targeted by the message is not an operation, the Compliance Checker must go through the various operations defined in that resource, to find one which the message complies with. A Protocol Fault (Table 6) is returned if none is found;
- Once the target operation has been identified, a thread is created in the silcode interpreter to execute that operation's code, produced previously by the SIL compiler before the resource has been registered in the Directory. That code can access other resources, according to what has been programmed in the operation;
- If the operation needs to send a message to another resource, as illustrated by Program 2, it has to pass the resource to send as a message to the Resource Serializer, which is done by the send operators (`<--` or `<||`, according to whether the message is synchronous or asynchronous, respectively).

6.3 Migration Path

SIL does not require a big bang migration path. An incremental e evolutionary approach can be achieved by using coexistence of several interoperability technologies and protocols.

SIL is application server and protocol agnostic and can coexist with SOA and REST applications. For example, the Jetty server used in the implementation of the SIL platform maintains normal HTTP capability, which means that it can also deal with SOA and REST based applications, by automatically choosing handlers based on the format of incoming messages.

The SIL server itself is able to deal with XML and JSON media types, through the stream concept. When a message is received, a set of available handlers are invoked to check whether they can process that message. The SIL message handler can be first and quick to recognize that the message is not SIL (lack of the right preamble), in which case it can invoke other handlers.

7 Conclusions and Future Work

Simplicity is the key concept in the IoT, in particular in what concerns the lower level devices. This has been the driving force behind the popularity of REST as service interoperability model, JSON as a serialization format and plain HTTP as a message level protocol. Web Services, SOA and XML can be too complex for simple applications, but offer the design time support that lacks in REST, JSON and HTTP, especially with simple devices that cannot cater for higher levels of interoperability. Application dynamicity is important, but so is verifiability and developer support.

REST APIs are simpler, but lack this design time support. It is also a fact that REST APIs in the context of IoT applications are extremely simple, with almost no states involved. This simplicity stems from applications, not from the technology, which was not conceived for the IoT, but rather for the Web, and has limitations, namely regarding binary support and asynchronous event processing.

Our opinion is that current Web technologies need to be reevaluated, taking into account the smaller granularity, higher constraints and even higher massive scale of networked devices. This is already happening, with adaptations of IPv6 to low power devices and sensor networks [22], as well as adaptations of HTTP for constrained resources [27]. A high legacy load is present, without achieving transparent compatibility. The design of a fresh solution, incorporating lessons learned and without being hampered by compatibility tradeoffs, has been the basic motivation for the design of the alternative solution that we have described in this chapter, which meant adopting the following main principles:

- Do not use text with markup for the resource serialization format, which is complex for human reading and inefficient for machine processing. Use two formats instead: programming language style for humans and binary for machine processing, with a compiler to link the two;
- Use complete separation of data and metadata (which text markup does not allow). This supports automatic use of metadata, only when needed. When the schema does not change, send only binary data, with design time checks done by the compiler;
- Use a message protocol with native support for binary data and asynchronous messages;
- Do not base schema level interoperability on schema sharing (as usually done in XML documents) or implied schemas (often, the case of JSON data). Use structural interoperability [37] instead, which decreases coupling;
- Do not use a separate language to describe schemas. Derive them automatically from the textual resource descriptions;
- Support a variable number of operations for each resource but support external access to resource structure as well (both measures contribute to a low modeling semantic gap);
- Support multinetwork resource references, so that heterogeneous networks can be seamlessly integrated.

A preliminary implementation of these principles exists, with a compiler and an execution platform for a language that implements these principles, SIL, but much remains to be done. In particular, the following aspects are already being tackled:

- Completion and optimization of the SIL platform;
- Extension of SIL to the upper semantic levels;
- A study comparing quantitative aspects (execution time and memory requirements) and qualitative aspects (ease of programming and of changing,

advantages and perils of platform independent code) between SOA, REST and SIL based solutions, especially in the low level granularity context of IoT;

- An assessment of scenarios of application. One particularly interesting concerns joining a SIL server and a conventional browser, working in tight cooperation, something we call the *browserver* [46] and that has been conceived to replace the browser as a Web access device (at the user's laptop, tablet or smart phone). This has the great advantage of turning the user into a first class Web citizen, improving the interactivity experienced by the user and allowing him to automatically offer services (including private information for personalization, context awareness, ambient intelligence, authentication, gathering statistics and information on usage patterns, direct browserver to browserver interaction, which can be used for group or collective intelligence, and so on).

References

1. Berners-Lee, T.: Weaving the web: the original design and ultimate destiny of the World Wide Web by its inventor. HarperCollins Publishers, New York (1999)
2. Luigi, A., Iera, A., Morabito, G.: The Internet of Things: A survey. Comput. Netw. 54, 2787–2805 (2010)
3. Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S.: Vision and challenges for realising the Internet of Things. Publications Office of the European Union, Luxemburg (2010)
4. Guinard, D., Trifa, V., Mattern, F., Wilde, E.: From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices. In: Uckelmann, D., Harrison, M., Michahelles, F. (eds.) Architecting the Internet of Things. Springer, Berlin (2011)
5. Abawajy, J.: Advances in pervasive computing. Internation. J. Pervasive Comput. Commun. 5(1), 4–8 (2009)
6. Guinard, D., Mueller, M., Pasquier-Rocha, J.: Giving RFID a REST: Building a Web-Enabled EPCIS. In: Proc. Second International Internet of Things Conference, pp. 1–8 (2010), doi:10.1109/IOT.2010.5678447
7. Earl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River (2005)
8. Richardson, L., Ruby, S.: RESTful Web Services. O'Reilly Media, Sebastopol (2007)
9. Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. "big"' web services: making the right architectural decision. In: Proc. International Conf. on World Wide Web, pp. 805–814. ACM Press (2008)
10. Mulligan, G., Gracanin, D.: A comparison of SOAP and REST implementations of a service based interaction independence middleware framework. In: Proc. Winter Simulation Conf., pp. 1423–1432. IEEE Computer Society Press (2009)
11. Becker, J., Matzner, M., Müller, O.: Comparing Architectural Styles for Service-Oriented Architectures - a REST vs. SOAP Case Study. In: Papadopoulos, G., et al. (eds.) Information Systems Development, pp. 207–215. Springer, US (2010)
12. Gupta, V., Udupi, P., Poursohi, A.: Early lessons from building Sensor.Network: an open data exchange for the web of things. In: Proc. Conf. on Pervasive Computing and Communications Workshops, pp. 738–744 (2010), doi:10.1109/PERCOMW.2010.5470530

13. Taherkordi, A., Eliassen, F., Romero, D., Rouvoy, R.: RESTful Service Development for Resource-Constrained Environments. In: Wilde, E., Pautasso, C. (eds.) REST: From Research to Practice. Springer Science+Business Media, New York (2011)
14. Guinard, D., Trifa, V., Wilde, E.: A resource oriented architecture for the Web of Things. In: Proc. Second International Internet of Things Conf., pp. 1–8 (2010), doi:10.1109/IOT.2010.5678452
15. Priyantha, N., Kansal, A., Goraczko, M., Zhao, F.: Tiny web services: design and implementation of interoperable and evolvable sensor networks. In: Proc. 6th ACM Conf. on Embedded Network Sensor Systems, pp. 253–266 (2008), doi:10.1145/1460412.1460438
16. Akribopoulos, O., Chatzigiannakis, I., Koninis, C., Theodoridis, E.: A Web Services-oriented Architecture for Integrating Small Programmable Objects in the Web of Things. In: Proc. Developments in E-systems Engineering Conf., pp. 70–75 (2010), doi:10.1109/DeSE.2010.19
17. Jammes, F., Mensch, A., Smit, H.: Service-Oriented Device Communications using the Devices Profile for Web Services. In: Proc. 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, pp. 1–8 (2005), doi:10.1145/1101480.1101496
18. Sakr, S.: XML compression techniques: A survey and comparison. J. Comput. Syst. Sci. 75(5), 303–322 (2009)
19. Moritz, G., Timmermann, D., Stoll, R., Golatowski, F.: Encoding and Compression for the Devices Profile for Web Services. In: Proc. 24th International Conf. on Advanced Information Networking and Applications Workshops, pp. 514–519 (2010), doi:10.1109/WAINA.2010.91
20. Sumaray, A., Makki, S.: A comparison of data serialization formats for optimal efficiency on a mobile platform. In: Proc. 6th International Conf. on Ubiquitous Information Management and Communication (2012), doi:10.1145/2184751.2184810
21. Hui, J., Culler, D.: IPv6 in Low-Power Wireless Networks. Proc. IEEE 98(11), 1865–1878 (2010)
22. Jacobsen, R., Toftegaard, T., Kjærgaard, J.: IP Connected Low Power Wireless Personal Area Networks in the Future Internet. In: Vidyarthi, D. (ed.) Technologies and Protocols for the Future of Internet Design: Reinventing the Web. IGI Global, Hershey (2012)
23. Hui, J., Thubert, P.: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. Internet Engineering Task Force (IETF) RFC 6282 (2011), `http://tools.ietf.org/html/rfc6282` (accessed April 30, 2012)
24. Shelby, Z., Bormann, C.: 6LoWPAN: The Wireless Embedded Internet. Wiley, UK (2009)
25. Lewis, G., Morris, E., Simanta, S., Wrage, L.: Why Standards Are Not Enough To Guarantee End-to-End Interoperability. In: Proc. Seventh International Conf. on Composition-Based Software Systems, pp. 164–173 (2008), doi:10.1109/ICCBSS.2008.25
26. Diallo, S., Tolk, A., Graff, J., Barraco, A.: Using the levels of conceptual interoperability model and model-based data engineering to develop a modular interoperability framework. In: Proc. Winter Simulation Conf., pp. 2571–2581 (2011), doi:10.1109/WSC.2011.6147965
27. Castellani, A., Gheda, M., Bui, N., Rossi, M., Zorzi, M.: Web Services for the Internet of Things through CoAP and EXI. In: Proc. International Conf. Communications Workshops, pp. 1–6 (2011), doi:10.1109/iccw.2011.5963563
28. Minoli, D.: Enterprise Architecture A to Z. Auerbach Publications, Boca Raton (2008)

29. Masse, M.: REST API Design Rulebook. O'Reilly Media, Sebastopol (2011)
30. Gislason, D.: Zigbee Wireless Networking. Elsevier, UK (2008)
31. Trifa, V., Wiel, S., Guinard, D., Bohnert, T.: Design and Implementation of a Gateway for Web-based Interaction and Management of Embedded Devices. In: Proc. 2nd International Workshop on Sensor Network Engineering (2009)
32. Fielding, R., Taylor, R.: Principled Design of the Modern Web Architecture. ACM Trans. Internet Technol. 2(2), 115–150 (2002)
33. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California at Irvine (2000), `http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf` (accessed April 30, 2012)
34. Appel, A., Jim, T.: Continuation-passing, closure-passing style. In: Proc. Symp. Princ. Program. Lang., pp. 293–302 (1989), doi:10.1.1.134.7735
35. Webber, J., Parastatidis, S., Robinson, I.: REST in Practice. O'Reilly Media, Sebastopol (2010)
36. Zyp, K., Court, G.: A JSON Media Type for Describing the Structure and Meaning of JSON Documents. Internet Engineering Task Force (IETF) draft-zyp-json-schema-03. (2011), `http://tools.ietf.org/html/draft-zyp-json-schema-03` (accessed April 30, 2012)
37. Delgado, J.: Structural interoperability as a basis for service adaptability. In: Ortiz, G., Cubo, J. (eds.) Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions. IGI Global, Hershey (2012)
38. Pautasso, C.: RESTful Web service composition with BPEL for REST. Data Knowl. Eng. 68(9), 851–866 (2009)
39. Guinard, D., Trifa, V., Pham, T., Liechti, O.: Towards physical mashups in the Web of Things. In: Proc. Sixth International Conf. Networked Sensing Systems, pp. 1–4 (2009), doi:10.1109/INSS.2009.5409925
40. Castellani, A., Bui, N., Casari, P., Rossi, M., Shelby, Z.: M Architecture and Protocols for the Internet of Things: A Case Study. In: Proc. International Conf. Pervasive Computing and Communications Workshops, pp. 678–683 (2010), doi:10.1109/PERCOMW.2010.5470520
41. Delgado, J.: Bridging the SOA and REST architectural styles. In: Ionita, A., Litoiu, M., Lewis, G. (eds.) Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments. IGI Global, Hershey (2012)
42. Dubuisson, O.: ASN.1 Communication Between Heterogeneous Systems. Academic Press, San Diego (2000)
43. Parr, T.: The Definitive ANTLR Reference. The Pragmatic Bookshelf, Raleigh (2007)
44. Lubbers, P., Albers, B., Salim, F.: Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. Apress, New York (2010)
45. Formica, A.: Similarity of XML-Schema Elements: A Structural and Information Content Approach. Comp. J. 51(2), 240–254 (2007)
46. Delgado, J.: The User as a Service. In: Vidyarthi, D. (ed.) Technologies and Protocols for the Future of Internet Design: Reinventing the Web. IGI Global, Hershey (2012)
47. Maeda, K.: Performance Evaluation of Object Serialization Libraries in XML, JSON and Binary Formats. In: Proc. Second International Conference on Digital Information and Communication Technology and its Applications, pp. 177–182 (2011), doi:10.1109/DICTAP.2012.6215346

48. Kumaran, S., et al.: A RESTful Architecture for Service-Oriented Business Process Execution. In: International Conference on e-Business Engineering, pp. 197–204. IEEE Computer Society Press (2008)
49. Xu, X., Zhu, L., Kannengiesser, U., Liu, Y.: An Architectural Style for Process-Intensive Web Information Systems. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 534–547. Springer, Heidelberg (2010)
50. Erenkrantz, J., Gorlick, M., Suryanarayana, G., Taylor, R.: From representations to computations: the evolution of web architectures. In: 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp. 255–264. ACM Press (2007)
51. Queinnec, C.: Inverting back the inversion of control or, continuations versus page-centric programming. ACM SIGPLAN Not. 38(2), 57–64 (2003)

The Role of Ad Hoc Networks in the Internet of Things: A Case Scenario for Smart Environments

Daniel G. Reina, Sergio L. Toral, Federico Barrero,
Nik Bessis, and Eleana Asimakopoulou

Abstract. Internet of Things is an emerging area and it visions an inter-connected world full of physical as well as virtual objects, devices, processes and services capable of providing a different lens on how to link them via the Internet. While Internet of Things as part of the Future Internet has been described as a paradigm that mainly integrates and enables several technologies and communication solutions a notable interest is to define how current standard communication protocols could support the realization of the vision. Within this context, we offer a state-of-the-art review on ad hoc and wireless sensor networks, near field communications, radio frequency identification and routing protocols as a mean to describe their applicability towards an Internet of Things realization. We conclude by presenting a brief case scenario to describe a future smart environment and illustrate its possible model architecture.

1 Introduction

For many years, wired networks used to be the only way to connect computers to the Internet. During the last decades, wireless communications have changed inter-connectivity by enabling computers to communicate and also exchange information stored on them on a wireless fashion. While the Internet is formed primarily by inter-connecting homogeneous devices (i.e. computers) there have been recently several paradigms in networking such as mobile, grid and cloud computing which enabled a purposeful inter-connectivity between various semi-homogeneous devices such as computers, cameras, smart-phones, sensors and other instrumentation (i.e. satellites).

The latest vision is to enlarge the inter-connectivity between devices making possible the formation of pure heterogeneous networks and contexts by inter-connecting hardware devices ranging from computers to simple sensors. This is by

Daniel G. Reina · Sergio L. Toral · Federico Barrero
Electronic Engineering Department, University of Seville, Spain
e-mail: dgutierrezreina@us.es, {toral,fbarrero}@esi.us.es

Nik Bessis · Eleana Asimakopoulou
School of Computing and Maths, University of Derby, UK
e-mail: n.bessis@derby.ac.uk, eleana.asimakopoulou@gmail.com

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 89–113.
DOI: 10.1007/978-3-642-34952-2_4

visioning an Internet of Things (IoT), an inter-connected world full of physical as well as virtual devices capable of providing services over the Internet. Within the IoT context, a thing refers to a physical or virtual object such as spaces and rooms, chairs, fruits, bottles, clothes, suitcases and bags, animals or even a process and a service like a cloud virtual machine.

During the last years, IoT has emerged as one of the most important paradigmatic strings of thought with regards of the future state of Internet. Its importance is described in terms of providing a different lens on how to link the Internet with real world's objects. In contrast to currently dominant paradigm within Internet which is based on human-to-human interaction, the IoT paradigm proposes a novel emerging paradigm of thought which postulates that any object, identified with a unique identifier will be considered as inter-connected [1]. As such, IoT has been proposed as a combination of the Internet and emerging technologies such as near-field communications, real-time localization, and embedded sensors as a way to transform everyday objects into smart objects [2]. Those objects can be transformed in ways that they can be understood better by reacting to and with their environment in a more advanced and meaningful manner. IoT has also been described as a paradigm that mainly integrates and enables several technologies and communication solutions including but not limited to tracking technologies, wired, wireless sensors, their networks, exchanged networked communication which in turn, lead to a shared next generation Internet, what is also known as Future Internet. IoT has also been defined as ''a world-wide network of inter-connected objects uniquely addressable, based on standard communication protocols." In a more comprehensive way, it has been perceived as a paradigm that connects real world with digital world [3].

Within this context, one of the fundamental challenges for the IoT realization is that like when integrating heterogeneous data that have been originally produced for a purpose other than their integration [4], objects also differ significantly in terms of their functionality, technology and application and in other words, they have been originally produced for a purpose other than their inter-connection over the Internet communication environment.

The development in digital hardware made possible portable computers, increasing the mobility, processing capability and reducing size and cost. While static powerful computers are already capable of participating in Internet and thus, in web-based communication services, small simple hardware devices will also be able to inter-connect in an IoT setting by using Radio Frequency Identification (RFID) techniques.

On the other hand, ad hoc networks have attracted a lot of attention in the last decades. They represent a new paradigm of communications where decentralized wireless nodes communicate with each other in a collaborative way to achieve a common goal. Nodes collaborate to establish unicast or multicast communications between a source node and a one or several destination node(s). When mobility of nodes is considered, communications refer to Mobile Ad Hoc Networks (MANETs). With the increment of mobile devices which are equipped with wireless transceivers such as smart phones, tablets, sensors and so on, the number of deployed devices with wireless communications capabilities is continuously increasing. Commercial wireless technologies such as Bluetooth, UWB, WiMAX, Wi-Fi or Zigbee make possible the connections among devices that are made by different manufactures, enabling ad hoc communications to be established on either regular or ad hoc basis. When vehicles are capable of exchanging information among them Vehicular Ad Hoc Networks (VANETs) are formed. Mobility is an intrinsic characteristic in VANETs, but unlike MANETs fixed mobility patterns are followed in vehicular scenarios. VANETs enable the formation of Intelligent Transport Systems (ITS). Normally, two types of communications can be found in ITS, (a) Vehicle-to-Vehicle communications (V2V) that is two or more vehicles forming a VANET, and (b) Vehicle-to-Infrastructure communications (V2I) that is a hybrid VANET with both static and mobile nodes. In general, the aforementioned communications can be extended to include nodes to infrastructure communications (N2I), where the nodes may be either vehicles or people. The fixed infrastructure can be easily connected to Internet acting as an access point for the VANETs or MANETs. Furthermore, the deployment of Wireless Sensor Networks (WSNs) is a reality in urban scenarios by sensing data parameters such as temperature, humidity, CO2 emissions, etc. The integration of MANETs, VANETs, WSNs and the fixed infrastructure is an interesting challenge which will enable the IoT manifestation, see Figure 1.

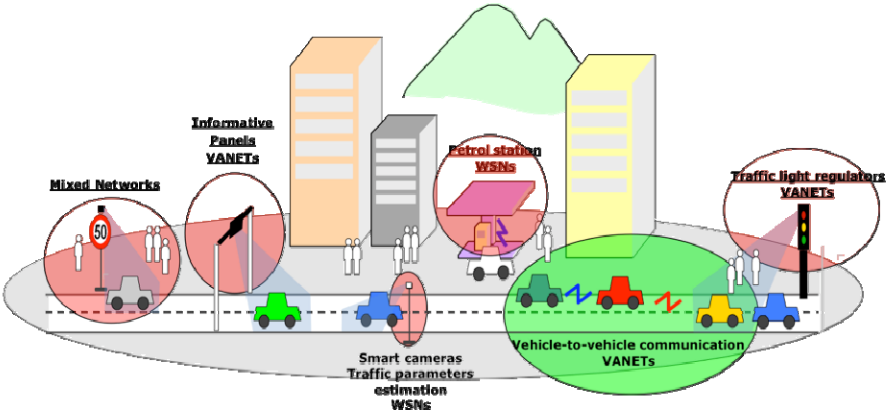

Informative Panels VANETs
Mixed Networks
50
Petrol station WSNs
Traffic light regulators VANETs
Smart cameras Traffic parameters estimation WSNs
Vehicle-to-vehicle communication VANETs

Fig. 1 Example of different network's deployments in the Internet of Things (IoT)

With these intelligent ambiences the provided services in wireless networks will increase notably as well as the complexity of them. The interactions with an urban smart environment will permit the collection of information from the environment and improve the decision-making of human beings. For that to happen, a high connectivity level between objects, processes, services and people must be guaranteed. While there will be a significant increase of the number of deployed wireless devices within the environment there should be a scalable infrastructure capable in enabling sufficient and full utilization of available resources as to take advantage of the IoT concept potential capabilities. Apart from that, the concept of green wireless networks has been lately appeared with the aim of reducing the use of resources in wireless communications as to reduce energy consumption. Having said that, it has been proved that the majority of energy is consumed during the access point stage in wireless communications since the terminal devices like mobile phones are optimized to be fed by batteries having low power consumption. The ad hoc networks have been proposed and implemented in numerous applications areas such as disaster management [5] [6] [7], health care [8], intelligent transportation systems [9], traffic management, and military applications among others [10], due to their self-organized and decentralized features.

In this chapter, we aim to offer a state-of-the-art review of the role of ad hoc networks in IoT. To achieve this, we start off with a review on the classification of ad hoc networks including mobile and vehicular ad hoc networks, wireless sensor networks and radio frequency identification. While we provide a discussion of their functionality we also highlight and brief their application and how these could be realized in an IoT setting (section 2). We also provide a discussion of routing protocols for IoT in an effort to present existing routing protocols applicability and suitability for an IoT realization (section 3). In section 4, we do present a visionary business scenario to illustrate a possible IoT model architecture. We finally conclude in section 5.

2 Classification of Ad Hoc Networks

2.1 Mobile Ad Hoc Networks (MANETs)

Mobile Ad Hoc Networks are self-organized networks which are deployed without the need for any fixed infrastructure. Having seen them as a new paradigm of mobile wireless communication, MANETs have attracted a lot of attention during the recent years. In MANETs every entity is called a node and works independently as a router. In the context of IoT, MANETs could represent scenarios such as people using mobile phones, a rescue team in an evacuation operation or soldiers in military applications, among others. MANETs are self-configuring, self-maintaining, self-healing, and self-repairing networks and such features are very suitable for mobile computing. The mobility of nodes is an intrinsic characteristic of nodes in MANETs which make even more challenging the deployment of these networks in real environments. The design of MANETs is much focused on routing protocols. They are one of the key components of MANETs. Figure 2 shows the importance of routing protocols in MANETs. The source node requires certain service A so it generates a discovery process to find such a service. The black arrows represent the discovery process flow. The intermediate nodes retransmit the incoming request until any request reaches the destination node. The destination node is the element of the network that can supply the required service.

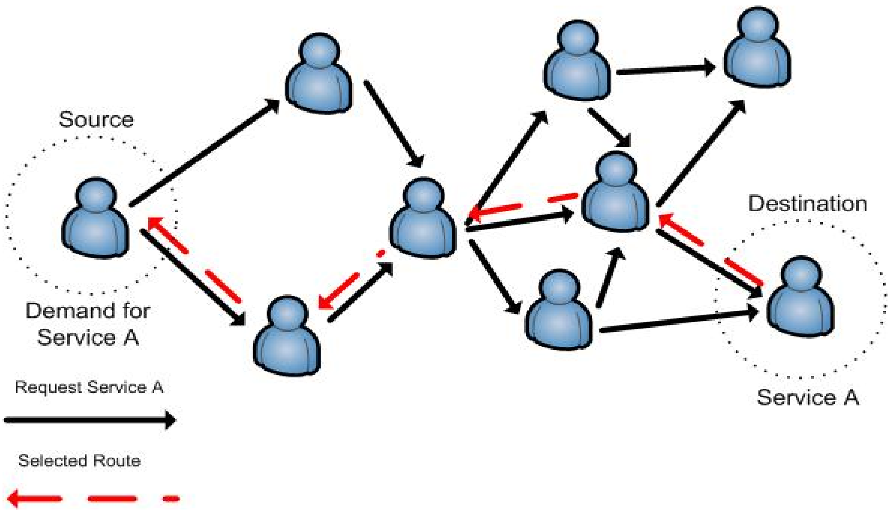


Fig. 2 Routing protocols in MANETs

Whenever several routes are found, the routing protocols are responsible for selecting the most appropriate route among those found. Several metrics are normally used to determine the quality of routes, such as hop count distance, end-to-end delay, and throughput. Since the mobility of nodes causes very changeable topologies, routing protocols should deal with such mobility conditions by acting against possible changes and implementing mechanisms to re-establish broken communication routes. Another important issue related to the discovery process of routing protocols in MANETs is the broadcast storm problem caused by the redundancy of request packets. As can be seen in Figure 2 many packets are redundant. This causes packet collisions and packet contentions which deteriorate the performance of ad hoc communications. In order to cope with this issue, several solutions have been proposed including GOSSIP, Multipoint Relay, Connected Dominant Sets and counter-based schemes. The main idea behind these algorithms is to reduce the number of redundant packets in the discovery process of routing protocols.

Mesh networks have appeared in recent years as an extension of typical ad hoc networks. Bruno et al [11], defined mesh networks as a flexible and low cost extension of wired infrastructure networks in which nodes collaborate with fixed infrastructure. Unlike MANETs, mesh networks are hierarchical networks, see Figure 3. Mobile nodes communicate with wireless routers which connect to access point in order to establish Internet connections. The wireless routers are forming a backbone which connects the “wired world” to the “wireless world”. Note that there is a high redundancy of connections in mesh networks so routing protocols must be focused on selecting the best path towards the wired world. Another important issue is to guarantee fairness in the network. MAC and routing protocols must guarantee that each user receives the same fair share of resources independent of how far is from the access point.

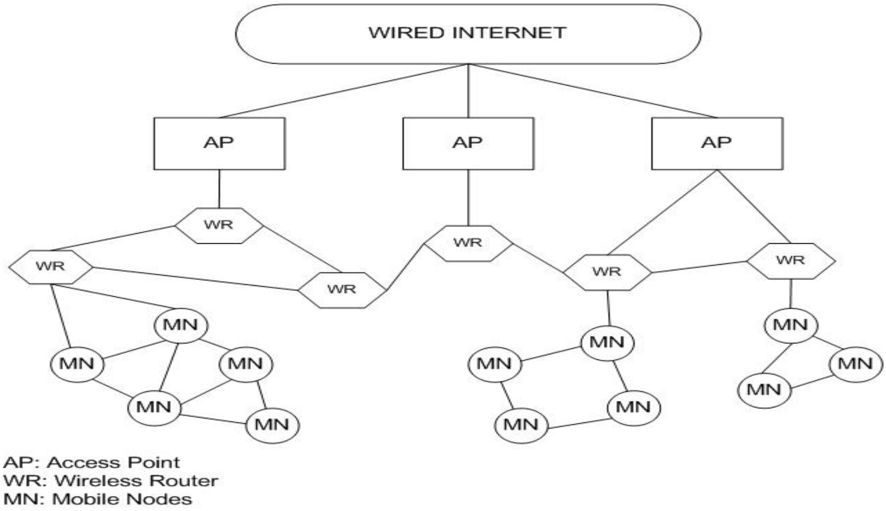


Fig. 3 Architecture of mesh networks

2.1.1 Service and Resource Discovery in Ad Hoc Networks

Service and resource discovery are also crucial for an efficient performance in ad hoc networks. Nodes must be aware of the available services and resources in their vicinity. Service and resource discovery mechanisms should work in collaboration with routing protocols. Two types of architectures have been proposed to develop service discovery [12]: 1) directory based architecture and 2) directory-less based architecture. The directory based architecture can also be divided into two categories: a) centralized directory and b) distributed directory. In the directory-less based architectures, nodes reactively request services and proactively advertise services. On the contrary, directory based schemes encompass a directory agent which is in charge of registering and handling services. Depending on the number of nodes which implement the directory agent we can distinguish between centralized directory and distributed directory. These nodes are responsible for keeping up-to-date the existing directory of services available in the network. The services discovery mechanisms are also very important for connecting ad hoc networks to the IoT.

2.1.2 Applications of MANETs

Since mobile ad hoc networks are self-organizing networks, they are suitable for those applications in which the deployment of a new fixed infrastructure is unfeasible and/or costly. In addition, the capability of dealing with mobility conditions makes MANETs appropriate for mobility applications. In addition, MANETs can also be used as a backup network whenever the main wired network fails, e.g. in disaster scenarios. The main applications of MANETs are [12]:

- Tactical networks: Military operations in battlefields
- Emergency services: Evacuating and rescue operations, disaster recovery, and health care applications
- Commercial and civilian environments: E-commerce, sport stadiums, and vehicular services among others
- Home and enterprise networking: Home networking, conferences, etc
- Education: Universities and virtual campuses
- Personal Area Networks: Clothing, etc
- Entertainment: Multi-user games, robotic pets
- Context aware services: Location specific services and time dependent services
- Coverage extension: Extending cellular network access.

2.1.3 Connecting MANETs to IoT

Several approaches have been proposed to connect mobile ad hoc networks to Internet. Since nodes in mobile ad hoc networks have IP addresses for routing purposes, it could be logical that such IPs may be used to route a packet through Internet. However, the main problem of this approach is that a node needs an efficient way to work out whether a certain address in the MANET is present or not and whether it is necessary to use a gateway or an access point. In principle, nodes are not aware of their contexts so it is difficult to collect neighboring nodes IPs. Discovery procedures must be carried out in order to collect neighboring information. However, these processes are normally time and message consuming since they require nodes to exchange a high number of packets. Normally, an access point should be placed so as to enable mobiles nodes connect to Internet. The effective placement of a gateway could be a challenging design factor due to the mobility of nodes and the optimum placement for a gateway could strongly depend on mobility conditions. As a consequence, the access point could be also mobile. Another approach is to use two different IPs, one to communicate through Internet and another one to identify nodes in the MANET. However, nodes can move freely so the target gateway could be changeable. If a node switches to another gateway, a new IP address should be used and the outgoing connections will probably break. Another possible approach is to use dynamic addresses by using the dynamic host configuration protocol (DCHP). This approach solves the problem of IP address when nodes are moving. On the other hand, the increasing use of smart mobile phones enable nodes to connect to Internet through cellular technologies such as 3G and 4G technologies, for instance the emerging Long Term Evolution (LTE) technology. However, these technologies are not unlicensed so users (or object owners within the IoT context) have to subscribe to these services. In addition, satellite communication can also be used in safety-related applications like military applications. To sum-up, the connection of ad hoc network to Internet is still a challenge requiring further research.

2.2 Vehicular Ad Hoc Networks (VANETs)

Vehicular Ad Hoc Networks could be seen as a specific type of MANETs. However, it has become a different research field in the last few years. Although this fact is noticeable, it is also clear that both VANETs and MANETs share similar features such as multi-hop communications, changeable topologies, mobility and power transmission constrains. VANETs have arisen from the increased wireless communications in vehicles. Nowadays, most vehicles are equipped with Bluetooth transceivers so it can be seen as the standard for intra-vehicle communications. To establish V2V other technologies such as WiFi and Zigbee are preferred since their performances are more suitable for longer distances. In recent years, the IEEE 1609 family for Wireless Access in Vehicular Networks (WAVE) [13] – which relies on the standard IEEE 802.11p – has become a promising technology for both V2V and V2I communications.

Figure 4 illustrates V2V communications in a VANET. This situation emulates a significant situation where one vehicle is aware of certain warning. This warning may be information about traffic or environment related conditions. In such situation, the node must inform others about the warning so as for other vehicles to adapt their behaviors appropriately. This dissemination should be done as quickly and as effective as possible. Since the density of nodes could be high, there is a trade off between reducing redundancy and increasing reliability of packets. Furthermore, mobility of nodes is a crucial parameter is VANETs since it is normally higher than in MANET scenarios. Mobility should be taken into account by routing and MAC protocols to adapt their performances to such high mobility conditions. The last vehicle on the queue, see Figure 4, may require adapting its speed to match the collected information from other vehicles and infrastructure. The establishment of connections should be done rapidly to permit the information to be exchanged by vehicles in a short time (directly via V2V or indirectly via V2I). For instance, Bluetooth connections take a long time to be established, and therefore Bluetooth could not be suitable for this short of V2V communications.

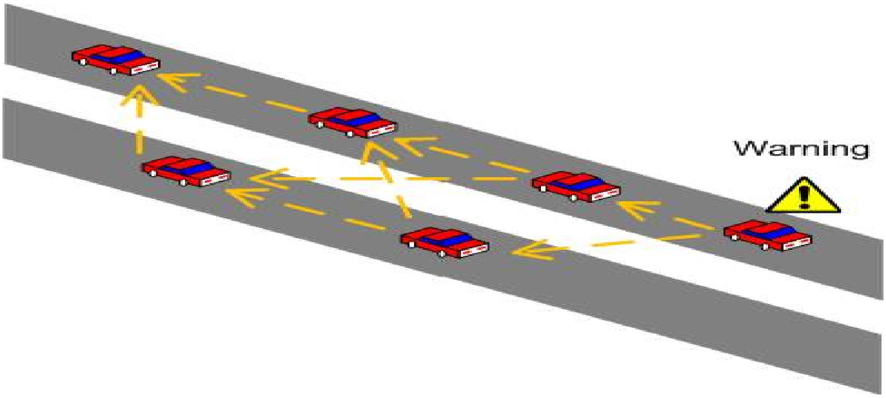


Fig. 4 Vehicle-to-Vehicle (V2V) communications

Several examples of V2I communications are illustrated in Figure 5. In these situations nodes are informed on certain conditions such as the state of traffic lights, traffic signals, or the state of traffic barriers. Clearly VANETs help to improve driver's decision making. Unlike pure MANETs, a VANET can collaborate with fixed infrastructure. The features of vehicular ad hoc networks can be improved by using wired network as backbone for providing data services. However, the deployment of Access Points (APs) is a challenging task since it depends on parameters like density and traffic conditions. The ideal placements for APs in vehicular networks are the typical vehicular public infrastructure such as traffic lights, light poles, and so on. Such hybrid behavior means that vehicles can communicate to APs within little number of hops leading vehicles forming self-organized wireless networks. A stand-alone sight of VANETs is only possible in dense networks. However, vehicular networks are very changeable and only under congested traffic flow such assumption could be ensured.

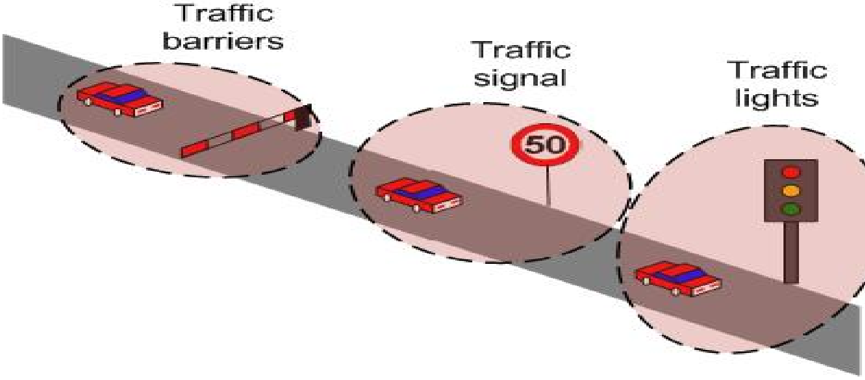


Fig. 5 Vehicle-to-Infrastructure (V2I) communications

Unlike mesh networks, VANETs are not hierarchical networks. In V2I communications, vehicles communicate directly with APs.

2.2.1 Applications of VANETs

Vehicular communications are aimed to form intelligent transportation systems using mobile devices and fixed infrastructure. Many applications are focused on improving the safety conditions in vehicles. The main applications of VANETs are [14]:

- Navigation safety applications: Prevention of traffic accidents, dissemination of warning messages, improvement of driver decision-making, post accident information, etc
- Navigation efficiency: Intelligent transportation systems, road congestion avoidance, and pollution mitigation among others

- Entertainment: Download multimedia, video streaming, etc
- Vehicle monitoring: Mobile sensor networks through vehicle communications
- Urban sensing: Congestion, traffic flows, pollution, etc
- Social networking: Friendship, proximity and correlation analysis
- Emergency: Evacuation emergency, disaster scenarios, etc.

2.2.2 Connecting VANETs to IoT

Similar mechanisms of that used in MANETs are also used in VANETs to connect vehicles to the IoT. Consequently, the vehicular networks are normally connected to Internet by means of APs using a Wireless Local Area Networks (WLAN) technology such as WiFi, WiMAx or Bluetooth.

2.3 Wireless Sensor Networks (WSNs)

Sensor networks are normally centralized networks where there is a central node in charge of gathering sensed data from sensor nodes [15]. The central node is called the sink of communications. The collected information is usually environment-related. Parameters such as temperature, humidity and proximity are normally measured. However, there have been important advances in electronic, micromechanical and chemistry manufacturing processes that make possible to find more sophisticated sensor nodes. The main characteristic of WSNs is the limited resources available in terms of memory and power energy. WSN nodes are fed by batteries so power consumption is an important design factor in WSNs. To tackle such constraints, nodes should transmit efficiently their sensed data to the sink node. Consequently, the majority of routing and MAC protocols for WSNs are focused on reducing the node's power consumption in order to extend the lifetime of the network and to avoid frequent battery replacements. The topologies of WSNs are less changeable than that of MANETs or VANETs. In general, nodes are static in WSNs, so topological changes are due to bad performances of nodes mostly, i.e. low battery problems or medium access problems. Peer-to-peer (P2P, also known as mesh), Star and Tree topologies are common topologies found in deployed WSNs, see Figure 6.

In star topology the nodes are normally located at only one hop distance from the sink so redundant data can be collected from different sensors. The sink is in charge of post-processing such information. In both mesh and tree topologies multi-hops communications take place. Several algorithms based on graph theory have been proposed to reduce power consumption such as minimum Connected Dominant Set (CDS) or minimum Spanning Tree. As the data is post-processed by sinks, they are normally connected to a higher-level network like Internet in order to monitor the network. With regard to IoT, WSNs can be seen in two different ways: 1) Every node is a different entity or 2) the whole network is an entity accessible through the sink node which has full information about the network. This point of view is very interesting since a WSN can be integrated into more complex networks. A further step in WSNs is the Wireless Body Area Networks (WBAN)

[16]. These networks rely on the feasibility of attaching or implanting very small bio-sensors inside the human body that are comfortable and that do not impair normal activities. The main application area of WBANs is health care applications. For example, nodes are attached on or inside human body in order to sense body parameters and then to communicate wirelessly to a central node which is connected to Internet. WBAN will effectively make possible, doctors to monitor patient's health in real time, anywhere and at any time.

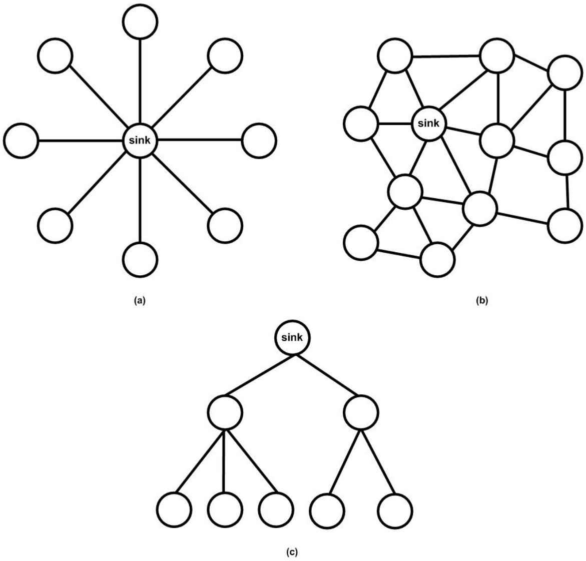


Fig. 6 WSN Topologies: (a) Star topology; (b) P2P or mesh topology; (c) Tree topology

The manufacturing process is the main challenge in WBANs since it is a multidisciplinary process involving electronic, chemistry and wireless communications, among others. There has also appeared a new tendency for including actuator nodes in WSNs forming a new type of network called Wireless Sensor and Actuator Networks (WSANs) [17]. In WSANs three types of nodes can be distinguished sink, sensor and actuator. While sensors are capable of sensing the environment, actuators are capable of acting on it. As in WSNs, sink nodes gather information from sensor nodes. WSANs should not be seen as a mere extension of WSNs since they have their own features. Actuator nodes are more complex and powerful nodes as compared to sensor nodes so a WSAN should not be considered as a homogeneous network. With regard to communication flows, there is a significant difference from WSNs. In WSANs multiple sensors may send data to a sink node, and multiple sinks may send data to an actuator node. As a consequence, communications can be divided into two types: one-to-many and many-to-one communications. To sum up, the interaction of WSNs with the IoT will enable to provide more useful services related to real-time data monitoring.

2.3.1 Applications of WSNs

The main applications of wireless sensor networks are related to monitoring ambient conditions. With the development of micro-electromechanical systems (MEMS) and digital electronic manufacturing, the variety of available sensors is increasing. In addition, the cost of sensor is decreasing as well. Such scenario makes possible to extend the scope of WSNs applications. The following list includes some important WSNs application areas [15] [18]:

- Military applications: Monitoring friendly forces, equipment and ammunition, battlefield surveillance, reconnaissance of opposing forces and terrain, and battle damage assessment among others
- Environmental applications: Forest fire detection, bio-complexity mapping of the environment, agriculture, flood detection, etc
- Healthcare applications: Tele-monitoring of human physiological data, tracking and monitoring doctors and patients and drug administration in hospitals
- Home applications: Home automation and smart environments.

2.3.2 Connecting WSNs to IoT

Since sensor nodes are simple devices with limited resources, the major issue is how to connect such simple devices to an inter-connected world of things. Several architectures have been proposed to connect WSNs to Internet. These architectures can be classified into three categories [19]: 1) the IP overlay over WSN, 2) the sensor overlay over IP, and 3) the higher-level gateway overlays.

When IP overlay over WSN, sensor nodes should be addressed with IPs as the same nodes connected to Internet. This scheme is complicated due to the limited networks resources of sensor nodes. In directed diffusion routing, which is typically used in WSNs, sensor nodes are not addressable with IPs. However, this model is drawing a lot attention in last few years thanks to the integration of IPv6 in sensor networks. In the second model data is encapsulated in IP packets. When the sensed data arrive at the sink, it encapsulates the data in IP packets. In the third level, WSNs and Internet are seen as two separate networks. A linking entity, the gateway, is responsible for adapting the incoming traffic from both networks. The gateway translates routing information of Internet into WSN routing mechanisms. Clearly, the first model represents the pure paradigm of the IoT in which each entity can be addressable. The protocol 6LoWPAN is an example of an implementation of the first model [20]. This is a version of the new IPv6 networking protocol for WSNs developed by the Internet Engineering Task Force (IETF) in the working group 6LoWPAN. The protocol 6LoWPAN is developed over the IEEE 802.15.4 standard. In this protocol, the features of IPv6 are adapted to the WSN constraints. The project Blip 2.0 [21], which is developed at the University of Berkeley, implements Ipv6 for TinyOS an operating system for WSNs [22]. In the second model, the sink node is connected to Internet and the sensor nodes are virtualized. The third model is the classical architecture for connecting WSNs to Internet. The sink node has an adapter in order to translate IP packets from Internet nodes.

On the other hand, the developments of middleware for WSNs are playing an important role in the introduction of WSNs in the IoT. Middleware provides users an abstraction of low communication layers of sensor nodes. MIRES is an example of middleware architecture for WSNs [23]. It is built on top of TinyOS and provides routing and service interfaces based on the publish/subscribe paradigm. WSN-SOA is an implementation of service-oriented architecture (SOA) for WSNs [24]. WSN-SOA is also implemented over TinyOS. The available attributes and the operations are described as web services. However, important modifications have to be done over the classical SOA in order to deal with the limited resources of wireless sensor nodes.

Constraint Application protocol (CoAP) [25], which is being developed by the IETF in the working group CoRE, is intended for designing a generic web protocol for the special requirements of this constrained environment, especially considering energy, building automation and other Machine-to-Machine (M2M) applications. CoAP is based on 6LoWPAN so it implements the first model. The interaction model of CoAP is similar to the client/server model of Hypertext Transfer Protocol (HTTP). Another similar approach was proposed in [26], the TinyREST that is a protocol aimed to connect WSNs to the Internet using Client/Server architecture. Sensor nodes in REST are addressed via Uniform Resource Locators (URL) using the Hypertext Transfer Protocol (HTTP) and its methods for accessing them. In TinyREST the client is a sensor and the server is a computer connected to the Internet. Consequently, TinyREST is based on the third connection model since there is a gateway to connect the WSN to the Internet. The HTTP methods such as GET, POST, PUT and DELETE, are also used in TinyREST. Moreover, TinyDB allows users to see WSNs as a database [27]. The data sensed by nodes is the information available in a database and it is accessible by sending SQL-like queries. Figure 7 illustrates several architectures for connecting WSNs to IoT.

On the other hand, Pachube is an open source platform that enables developers to connect sensor data to the IoT [28]. Pachube lets user tag and share data from physical and virtual devices through the Internet. The goal of Pachube is connecting to the environment rather than connecting to things. Pachube platform allows users to visualize world-wide data sensor through the Internet.

Furthermore, the idea of including sensor networks in the IoT is attracting the attention of several large companies. For example, the Hewlett-Packard with the Central Nervous System for the Earth (CeNSE) project is aimed to build a worldwide sensor network. The main goal of CenSE project is to deploy a massive amount of nano-scale sensors and actuators embedded in the environment and connect them via an array of networks with computing systems, software and services to exchange their information among analysis engines, storage systems and end-users. The main feature of CeNSE project is that HP is developing its own technology based on accelerometer to measure environmental parameters.

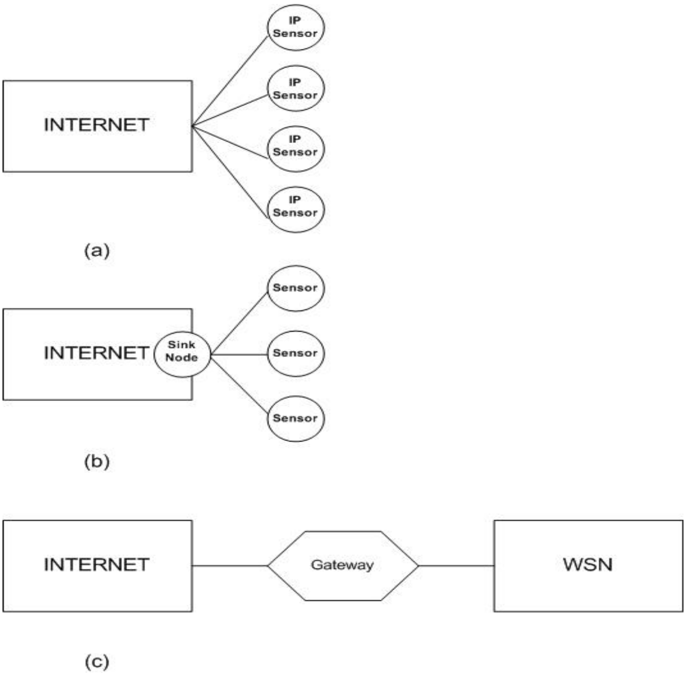


Fig. 7 Architectures for connecting WSNs to IoT

2.4 Radio Frequency Identification (RFID)

Another promising technology for supporting the IoT manifestation is RFID technology. It enables low-data communications between a simple device so called a tag, and a tag reader which is normally connected to a computer system. RFID communications have been used to identify and track objects wirelessly. Unlike bar codes, the RFID tags do not need to be within line of sight to communicate each other.

Two types of tags can be found, 1) passive tags which do not rely on any energy source and 2) active tags which contain an energy source like a battery. The main advantage of passive tags is that they do not require any power supply so they are simpler and cheaper than active tags. Passive tags use the radio energy transmitted by the reader as its energy source. The low cost of passive tags will enable a massive deployment of RFID tags attached to ordinary things like clothes, suitcases, bags, and so on. The information stored in the tags depends on the target application and the storage capability of a tag is limited by a few kilobytes of data.

RFID systems currently operate in the Low Frequency (LF), High Frequency (HF) and Ultrahigh Frequency (UHF) bands. Each frequency has its advantages and disadvantages. There is not any ideal frequency for all applications. In general, lower frequency means lower read ranges and slower data read rates.

LF-RFID systems are typically 125 kHz with a shorter read range (<0.5 m or 1.5 ft). LF-RFID systems tend to be less sensitive to interference than higher RFID systems. HF-RFID systems operate at 13.56 Mhz with read range which less than 1 m or 3ft. UHF-RFID systems utilizes the 860 to 930 MHz band, typically 860 MHz in Europe and 930 MHz in North America. The read range of ultrahigh RFID systems is up to 3 m or 9.5 ft.

RFID technology has been envisioned as the key wireless technology to accomplish the IoT. For instance, Electronic Product Code (EPC), which was created at MIT Auto-ID Center, was conceived as the starting point to develop the IoT. The EPC was designed as a universal identifier of every physical object in the world. Currently EPC is managed by EPCGlobal and it is aimed to identify a specific item in a supply chain context. In addition to the EPC code, EPCGlobal also provides the necessary infrastructure for a global IoT. However, EPCGlobal objectives are focused on industrial applications and in particular, for serving tracking and logistics management. The EPC network architecture enables partners of a business chain to share information. The main functionality of EPC network can be summarized as follows [29]:

- Provide linkage between physical objects and EPC tags.
- Manage huge amount of data from RFID sources.
- Provide a universally data format for transferring information.

The EPC network architecture is composed of tags, readers, middleware layer, information service layer, Object Name Service (ONS), Discovery Service (DS) and the Enterprise Applications [30]. Tags and readers are the sources of information. The middleware layer so-called Savant is in charge of capturing information from readers and managing that in order to provide meaningful data. The information service layer acts as a repository about any items identified. The ONS allows tracking objects and the DS is a set of service that enables user to find the data related to specific objects. Further detail about EPC network can be found in [29]. On the other hand, eCloudRFID [30] is framework architecture for mobile devices with the goal of facilitating the development process of embedded RFID applications and the integration of business applications and EPC networks instances. As in EPC network architecture, a middleware layer is necessary to connect the physical world with the IoT.

The RFID ecosystem [31] created at the University of Washington is oriented to investigate patterns of adoption and utilization of RFID applications in a realistic day-to-day setting. They pointed out that creating RFID applications for IoT is challenging since the data associated with tags, antennas, and events must be personalized and carefully controlled to create a safe, meaningful and user experience. They developed several RFID-based web applications such as a search engine for things, social applications, a digital diary, and an event-based search. Such applications can be personalized by using a tag manager. This application enables to transform RFID data into high-level events. The results in [31] show that most users were interested in using RFID applications especially the digital diary.

2.4.1 Applications of RFID

The main applications of RFID communications are identification-related and tracking. However, during the last years new applications are emerging [32] such as:

- Access management
- Retailing industry
- Food and restaurant industry
- Health care industry
- Library applications
- Travel and tourism industry
- Toll collection and contactless payment
- Smart-dusts
- Mechanism to speed up the pairing phase of Bluetooth and WiFi communications (NFC)
- Social networking.

2.4.2 Connecting RFID to IoT

The RFID tags are so far the simplest objects that can be connected to the IoT. RFID readers can collect information from tags and make such information accessible to the Internet. The RFID readers act as translators. As a consequence, those mechanisms applied to WSNs can also be applied to RFID communications. The RFID readers act as sensor nodes and the RFID technology is the wireless interface used to collect information from the tags. The information stored in the tags represents the data sensed from the environment. However, unlike sensor networks, the measurements are triggered whenever a tag gets closer to a reader.

A centralized architecture is presented in Figure 8, in which each reader is connected to a server. This server connects the RFID reader to the Internet. This architecture was adopted in the RFID ecosystem [31].

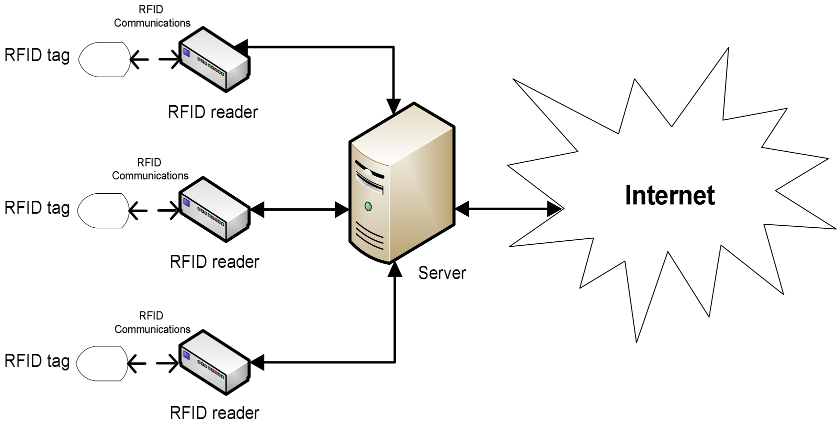


Fig. 8 Connecting RFID technology to IoT

In general, a middleware layer is needed in order to pass a request from the application to the readers. The main tasks of the middleware layer are data filtering and aggregation. Note that the amount of data from readers may be very large and redundant. Savant and eCloudRFID middleware are some example of middleware for connecting RFID data to the IoT. In [33] the authors proposed integrating IPv6 in RFID tags so whenever a RFID reader pass closer it can obtain an IPv6 address for connecting to the Internet.

Mobile phones can also be used as NFC readers so they can serve as translators, converting RFID data into Internet data, see Figure 9. Since the new generation of smart-phones incorporates the functionality for Internet connectivity, the approach can be a reality in the near future.

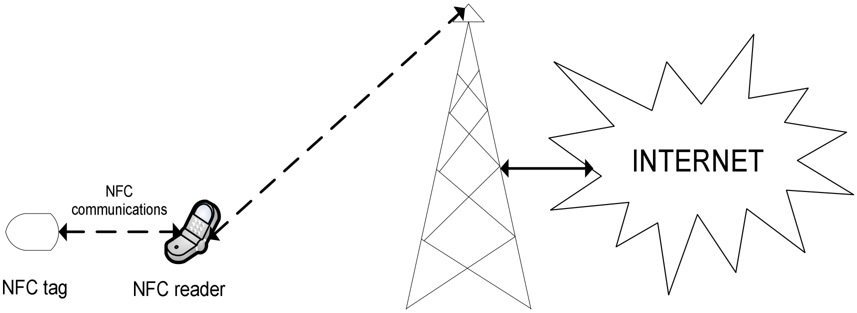


Fig. 9 Connecting NFC to IoT

2.5 *Near Field Communications (NFC)*

Near Field Communication (NFC) is a set of standards for short-range communications. NFC attracts much of attention [34] and it is estimated that by 2015 today's market value will be increased by eight times. [35] suggest that within the same time frame 785 million NFC enabled devices will be spread across the world, mainly incorporated within smart-phones.

Various companies already utilize NFC, based on the fact that users are more comfortable with using mobile devices as secure payment tools. McDonalds aims to expand the NFC potential by conducting trials that combine mobile-based coupon distribution with payments and collecting user data for marketing purposes [36]. Barclays Bank and Orange launched a service allowing their consumers to tap their phones in order to pay for purchases up to a specific amount by bringing the phone into a close proximity range. Starbucks allows customers to swipe their phones by using an internal service for making payments, instead of using cards or cash. Markets in France use NFC for improving the shopping experience for the visually impaired or elderly people [37].

In NFC communications there is an initiator and a target device that is normally a passive tag. The main advantage provided by NFC technology is that it has been incorporated in the new generations of mobile phones. The first mobile phone which used NFC was the Nokia 6216 classic. New smart phones, like Nexus S of Google, are also incorporating NFC capabilities. As a consequence, the mobile phone can act as the NFC initiator. Since current mobile phones are connected to Internet, they can easily serve as bridges to transfer RFID information to the IoT. The use of NFC will enable an expansion of RFID applications.

NFC operates at 13.56MHz and its data rates are ranging from 106 kbit/s to 424 kbit/s. NFC protocols cover communication protocols and data exchange formats. It includes RFID protocols such as ISO/IEC 14443 and FeliCa and other protocols such as ISO/IEC 18092 and those defined by NFC forum. This forum was founded by Nokia, Philips and Sony in 2004 and currently more than 150 companies have been incorporated.

3 Routing Protocols for the IoT

The development of routing protocols is a very active research field in ad hoc networks. The design of routing protocols for ad hoc networks is challenging due to mobility conditions and the limited resources of nodes. Most routing protocols for ad hoc networks are focused on guaranteeing Quality of Service (QoS) metrics such as bandwidth and end-to-end delay [38] [39]. On the other hand, routing protocols for WSNs are focused on maximizing network's lifetime by reducing the energy consumption [40]. However, the introduction of ad hoc networks in IoT requires new routing protocols oriented to connecting such limited devices to the Internet. Routing protocols for the IoT must guarantee connectivity, fairness and QoS between the nodes both in ad doc networks and the APs. Note that it is clearly different from the classical concept of routing protocol for ad hoc networks where QoS must be guaranteed between any pair of nodes in the network. In an the IoT setting, routing protocols must ensure fairness so that each node can communicate with the APs. Hierarchical solutions are normally adopted in order to reduce redundancy and for ensuring data association and data aggregation. Moreover, cross-layer designs are attracting attention since they are suitable for variable channel conditions which are normally found in ad hoc scenarios. Cross layers designs make possible the collaboration between MAC and routing layers so as to optimize routing decisions. A routing protocol using Received Signal Strength Indicator (RSSI) is an example of a cross layer design in which the routing protocol can use RSSI values to estimate Euclidean distance between two nodes or the link quality.

One possible solution is to adapt existing routing protocols to the requirements of the IoT. For example classical routing protocols for ad hoc networks such as Ad Hoc On-Demand Distance Vector (AODV) routing protocol and Dynamic Source Routing (DSR) can be modified to fulfill IoT requirements. For instance, AOMDV-IOT [41] is an extension of Ad Hoc On-demand Multipath Distance Vector routing protocol. AOMDV allows a user to find several node-disjoint routes and link-disjoint routes between a source node and a destination node. However, in an IoT context the objective is to find a node connected to the Internet. This issue is solved by AOMDV-IOT and in particular, by implementing an Internet Connecting Table (ICT). In addition, an Internet Linking Address (ILA) is also defined so as to be used when the node is connected to the Internet.

Mesh networks are focused on the creation of hierarchical architectures that enable mobile nodes to connect to the fixed infrastructure creating a mixed network. Clustering algorithm can be a solution to accomplish such associability. Mesh Adaptive Routing Tree (MART) [42] is defined by the IEEE 802.15.5 working group and its objective is to develop a routing tree for Mesh networks. In tree architecture, a node can only communicate with its one-hop neighbors. A Hierarchy structure is built in order to forward packets from the root to the leaves. Three phases are defined in MART: 1) initialization (or configuration) phase, 2) normal phase, and 3) recovery phase. During the initialization the tree is formed. The number of branches of each node depends on its capacity. The MART tree formation is functionally divided into two stages: association and address assigning. In the normal phase, packets can be routed throughout the tree. Finally, the recovery phase is carried out whenever broken links are detected.

With regard to WSNs, routing strategies are being focused on integrating IPv6 so that each node is the network that can be identified by an IP address. In this way, RPL routing protocol [43], which was developed by the IETF in the working group namely Routing Over Low power and Lossy networks (ROLL), is a distance vector based IPv6 routing protocol which specifies how to build a Destination Oriented Directed Acyclic Graph (DODAG) using an objective function and a set of metrics/constraints. These metrics determine the quality of the paths found. Depending on the requirements of the deployed application different metrics can be defined and multiple DODAGs can be defined in order to satisfy such requirements. Note that RPL is a hierarchical routing protocol. The graph starts at the root called LowPan Border Router (LBR). ICMPv6 messages are then exchanged by nodes in order to share graph related information. In DODAG formation, each node of the graph has to select a parent node (or multiple parents depending on the objective function) in a hop-by-hop fashion and the leaf nodes can communicate with the root node by just forwarding the packet to its immediate parent. In addition, RPL also supports P2P communications so any node can communicate with any other node in the network. On the other hand, existing routing protocols for WSNs can also be adapted to support IPv6. For instance, 6LoWPAN SPEED [44] is an evolution of SPEEP protocol. SPEED is a routing protocol that supports soft real-time communications in large-scale sensor networks. The end-to-end soft real-time is achieved by maintaining a desired delivery speed across the network by means of feedback control and non-deterministic geographic forwarding. Three types of communications services are implemented in SPEED routing protocol, 1) real-time unicast, 2) real-time area-multicast, and 3) real-time area-anycast. In SPEED protocol, each packet is forwarded towards the direction of the destination node. In [44] speed protocol is adapted to support 6LoWPAN by encapsulating SPEED messages into ICMPv6 headers. This mechanism based on encapsulation may be used by other routing protocols in WSNs.

4 Towards an IoT Smart Environment

However, a concern [45] with current real-world implementations is that they cover narrow visions where technology core specification stands in-between organizations and end-users as an instrument for data gathering.

In this chapter, we describe a "thereuGO" case scenario incorporating an all-in-one inclusive approach. That is by suggesting the use of the MANETS, RFID, NFC and IoT to transform physical and virtual business processes, services and products into smart objects and inter-connect them into an Internet-like structure. This is by tagging them in a way that customers and businesses can perform intelligence by using personalized technology and other computational approaches like Cloud computing to manage their tailored interactions in a scalable manner.

While the "thereuGO" case scenario is driven by the specifics of a gaming operator, there is evidence of its scope and applicability in wider business and organizational contexts.

4.1 The "thereuGO" Case Scenario

Bob is an occasional player in Gaming Operator (GO), one of Europe's leading gaming operators. When visiting GO, Bob spends few hours playing, and socializing with others like Alice and Ted. Bob owns a smart-phone, which effectively enables him to access Internet services through WiFi. When entering GO premises, Bob does not need to show his loyalty card, "thereuGO" (the acronym for GO's smart IT environment) registers his entry automatically. In fact, "thereuGO" informs Carol who is the manager and bar attendant – a few rooms away from Bob's positioning – to pour a pint of Guinness ready for Bob to collect next to his favorite slot machine. As Bob enters from one room to another, his smart-phone guides him to his favorite slot machine that is available at the time. Most importantly, it tells him what services and products are available in a dynamic and timely fashion for each room as he enters them. He finds the room-based browse feature very exciting; he is now aware of things that he had never noticed them before like the odd slot machine with the most money to be won; last winner was three weeks ago; Bob took a picture of it and shared it with Carol. This Friday, Bob decided to play cards. Using his smart-phone, he browses the tables available and realizes that Alice and Ted are also playing cards specifically, on table 3, room 3. He claims the space and as Carol tracks his way, Bob confirms delivery of his drink in table 3. Later on, Bob used his smart-phone to order some drinks by taking a photo of the label from the printed menu. While they were chatting, Bob informed that the little odd slot machine is now available but he decided to stay with his friends. Few minutes later, Alice coveted a plate of cold snack that she had never seen before when a waiter delivered it to the table next to them. She took a picture of it, checked the ingredients (Alice knows that by reading Ted's social network profile tat he is allergic in nuts) and sent the order to Carol. Minutes after, Ted reserved more drinks for later (all like Blue-Monkey by the reading of their social network sites); he found the offer sent from Carol a timely opportunity not to miss. Carol and her team now feel much more comfortable in responding to customers' preferences and ad-hoc requests and most importantly, they can now manage their resources more efficiently; they know – at anytime – what are the most and least desirable services; which ones are available; who is drinking what at what frequency; when they would most likely need top-up; what are their stock levels and profits; how many people in each room; how many in the pre-hallway entry for more than 5 minutes, etc. Carol knows that undecided newcomers will most likely accept a treat. All these, through a GUI which illustrates the relationship between the customers and her tagged smart environment resources. “thereuGO” seems to be a Win-Win and Show-me-the-money opportunity for customers and business; both ends can self-manage their desires and commodities. Later on, our actors waived each other and promised to play cards online. On the way out, Bob realizes that the odd slot machine still has the money; he is now thinking to play from home online.

4.2 A “thereuGO” IoT Model Architecture

Figure 10 extends low-level architectures discussed in [4] [5] [6] by illustrating more technical aspects related to the “thereugo” case scenario.

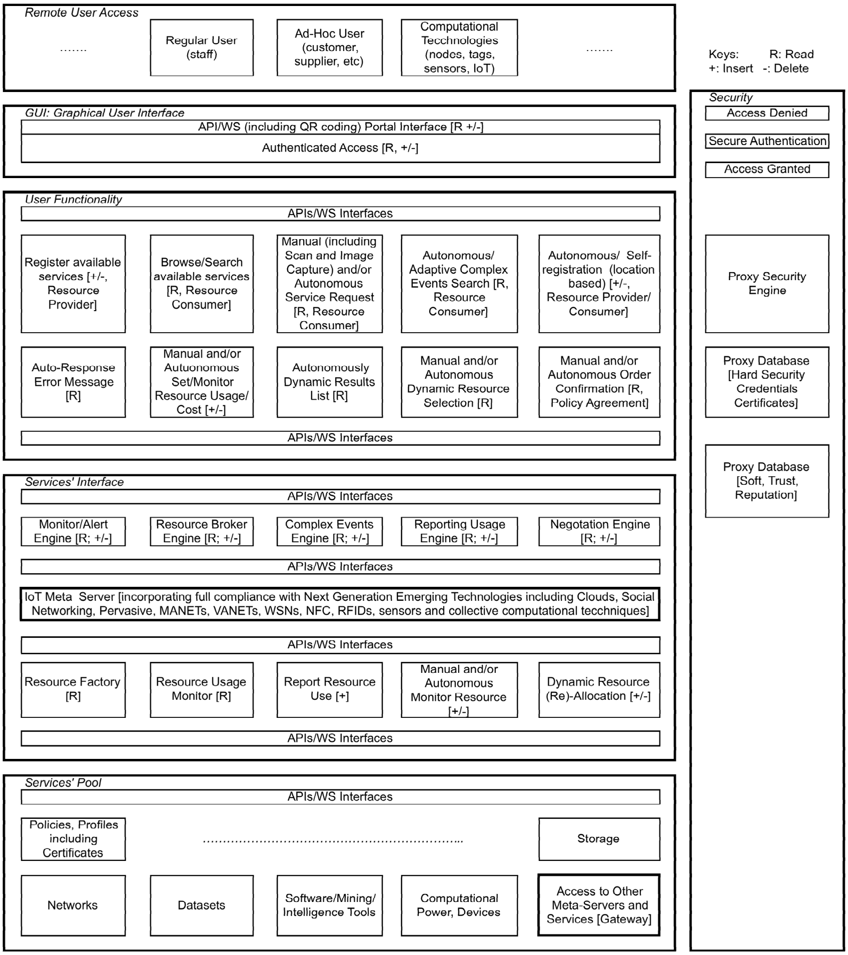

Remote User Access
....
Regular User (staff)
Ad-Hoc User (customer, supplier, etc)
Computational Tecchnologies (nodes, tags, sensors, IoT)
....
Keys: R: Read
+: Insert -: Delete
GUI: Graphical User Interface
API/WS (including QR coding) Portal Interface [R +/-]
Authenticated Access [R, +/-]
User Functionality
APIs/WS Interfaces
Register available services [+/-, Resource Provider]
Browse/Search available services [R, Resource Consumer]
Manual (including Scan and Image Capture) and/or Autonomous Service Request [R, Resource Consumer]
Autonomous/ Adaptive Complex Events Search [R, Resource Consumer]
Autonomous/ Self-registration (location based) [+/-, Resource Provider/ Consumer]
Auto-Response Error Message [R]
Manual and/or Autuonomous Set/Monitor Resource Usage/ Cost [+/-]
Autonomously Dynamic Results List [R]
Manual and/or Autonomous Dynamic Resource Selection [R]
Manual and/or Autonomous Order Confirmation [R, Policy Agreement]
APIs/WS Interfaces
Services' Interface
APIs/WS Interfaces
Monitor/Alert Engine [R; +/-]
Resource Broker Engine [R; +/-]
Complex Events Engine [R; +/-]
Reporting Usage Engine [R; +/-]
Negotation Engine [R; +/-]
APIs/WS Interfaces
IoT Meta Server [incorporating full compliance with Next Generation Emerging Technologies including Clouds, Social Networking, Pervasive, MANETs, VANETs, WSNs, NFC, RFIDs, sensors and collective computational tecchniques]
APIs/WS Interfaces
Resource Factory [R]
Resource Usage Monitor [R]
Report Resource Use [+]
Manual and/or Autonomous Monitor Resource [+/-]
Dynamic Resource (Re)-Allocation [+/-]
APIs/WS Interfaces
Services' Pool
APIs/WS Interfaces
Policies, Profiles including Certificates
....
Storage
Networks
Datasets
Software/Mining/ Intelligence Tools
Computational Power, Devices
Access to Other Meta-Servers and Services [Gateway]
Security
Access Denied
Secure Authentication
Access Granted
Proxy Security Engine
Proxy Database [Hard Security Credentials Certificates]
Proxy Database [Soft, Trust, Reputation]

Fig. 10 A IoT Technical Model Architecture

Our past works explain the flow of interactions between computational devices capable of sensing the environment and establishing an ad-hoc mobile network. Herein, we use the “thereuGO” case scenario to demonstrate how the functionality available and the aforementioned technologies relate and impact in realizing, making sense of and ultimately enabling a more informed decision-making based on the actual situation rather than a speculative analysis. Specifically, the model appreciates that each user may have different needs, which requires personalization technologies like personalized URLs and personalized femtocell techniques. Due to the complexity involved we have not made links between functions and services.

In terms of the functionality available, the model appreciates that users will use their smart devices to access resources available from the smart environment remotely and on an ad hoc basis. Users get access to the portal after a successful authentication control. Authentication takes decisions on the basis of both security standards (PKI, X509, etc) and softer issues such as privacy, trust and reputation as there is a need to ensure the reputation of a service requestor and/or provider. Following the authentication procedure users can register their resources using some metadata descriptions, which can be stored in a factory for their future harvesting. Users may also request for resources in either manual or autonomous manner. Following the search procedure (manual or autonomic) a broker will negotiate between resource provider and requestor on the basis of user profile and policies prior to any resource confirmation and allocation. A monitoring function is used to dynamically re-allocate resources when these become unavailable for any reason. A complex events engine is suggested in order to monitor and ensure that combination of tailored parameters may lead to alerts. It is important to note that each function or service support multiple instances regardless if they are shown as single instances.

5 Conclusions

In this chapter, we provided a state-of-the-art review on how current standard communication protocols could support the realization of the IoT vision. In particular, we discussed ad hoc and wireless sensor networks, near field communications, radio frequency identification and routing protocols as a mean to describe their applicability towards the IoT realization.

Within this context, we highlighted that although most standard communications and protocols are supportive their connection to Internet and thus, to the IoT is still a challenge which requires further research. We also presented a brief case scenario describing a future smart environment; this was to illustrate its possible IoT model technical architecture.

Our future work involves the identification of suitable network simulation environments; this will be of particular importance since the IoT will open several opportunities in the real-world. This study, will also aim to define the network performance and metrics for several IoT case scenarios.

References

1. Tan, L., Wang, N.: Future Internet: The Internet of Things. In: 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), August 20-22, pp. V5-376–V5-380 (2010)
2. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Computer Networks 54(15), 2787–2805 (2010)
3. Presser, M., Gluhak, A.: The Internet of Things: Connecting the Real World with the Digital World. In: EURESCOM mess@ge – The Magazine for Telecom Insiders, vol. 2 (2009)
4. Bessis, N., Asimakopoulou, E., French, T., Norrington, P., Xhafa, F.: The Big Picture, from Grids and Clouds to Crowds: A Data Collective Computational Intelligence Case Proposal for Managing Disasters. In: 5th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2010), November 4-6, pp. 351–356 (2010)
5. Asimakopoulou, E., Bessis, N., Varaganti, R., Norrington, P.: A Personalised Forest Fire Evacuation Data Grid Push Service – The FFED-GPS Approach. In: Asimakopoulou, E., Bessis, N. (eds.) Advanced ICTs for Disaster Management and Threat Detection: Collaborative and Distributed Frameworks, pp. 279–295. IGI (2010) ISBN: 978-1615209873
6. Bessis, N., Asimakopoulou, E., Xhafa, F.: A Next Generation Emerging Technologies Roadmap for enabling Collective Computational Intelligence in Disaster Management. International Journal of Space-Based and Situated Computing (IJSSC) 1(1), 76–85 (2011)
7. Reina, D.G., Toral, S.L., Barrero, F., Bessis, N., Asimakopolou, E.: Modelling and assessing ad hoc networks in disaster scenarios. Journal Ambient Intelligence and Humanized Computing, JAIHC (2012), doi:10.1007/s12652-012-0113-3.
8. Hande, A., Ersoy, C.: Wireless sensor networks for health care: A survey. Computer Networks 54, 2688–2710 (2010)
9. Gutiérrez-Reina, D., Toral, S.L., Johnson, P., Barrero, F.: An evolutionary computation approach for designing mobile ad hoc networks. Expert Systems with Applications 39, 6838–6845 (2012)
10. Wolfgang, K., Martin, M.: A survey on real-world implementations of mobile ad hoc networks. Ad Hoc Networks 5, 324–339 (2007)
11. Bruno, R., Conti, M., Gregori, E.: Mesh Networks: Commodity multihop ad hoc networks. IEEE Communications Magazine 3, 123–131 (2005)
12. Hoebeke, J., Moerman, I., Dhoedt, B., Demeester, P.: An overview of mobile ad hoc networks: Applications and chanllenges. Journal of Communications Networks 3, 60–66 (2004)
13. Morgan, Y.L.: Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics. IEEE Communications Surveys & Tutorials 12(4) (2010)
14. Gerla, M., Kleinrock, L.: Vehicular networks and the future of internet mobile. Computer Networks 55, 457–469 (2010)
15. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor networks survey. Computer Networks 52, 2292–2330 (2008)
16. Huasong, C., Leung, V., Chow, C., Chan, H.: Enabling technologies for wireless body area networks: A survey and outlook. IEEE Communications Magazine 47, 84–93 (2009)

17. Bur, K., Omiyi, P., Yang, Y.: Wireless sensor and actuator networks: Enabling the nervous system of active aircraft. IEEE Communications Magazine 48, 118–125 (2010)
18. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40, 104–112 (2002)
19. Cristin, D., Reinhardt, A., Parag, S., Mogre, P.S., Steinmetz, R.: Wireless sensor networks and the internet of things: Selected Challenges. Structural Health Monitoring 5970, 31–33 (2009)
20. Jeonggil, K., Terzis, A., Dwason-Haggerty, S., Culler, D.E., Hui, J.W., Levis, P.: Connecting low power and lossy networks to the internet, vol. 49, pp. 96–101 (2011)
21. http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip
22. http://www.tinyos.net/
23. Souto, E., Guimaraes, G., Vasconcelos, G., Vieira, M., Rosa, N., Ferraz, C., Kelner, J.: J. Personal and Ubiquitous Computing 10, 37–44 (2006)
24. Avilés-López, E., García-Macías, A.: TinySOA: A service oriented architecture for wireless sensor networks. Service Oriented Computing and Applications 3, 99–108 (2009)
25. Shelby, Z., Hartke, K., Bormann, C., Frank, B.: Constrained Application Protocol (CoAP). draft-ietf-core-coap-09 (2012)
26. Luchkenbach, T., Gober, P., Arbanowski, S., Kotsopoulos, A., Kim, K.: TinyREST- a protocol for integrating sensor networks into the internet. In: Proceedings of REALWSN (2005)
27. http://telegraph.cs.berkeley.edu/tinydb/
28. https://pachube.com/
29. Leong, K.S., Ng, M.L., Engels, D.W.: EPC network architecture. Autoidlabs-wp-swnet-012, 1 edn. white paper series, pp. 1–14 (2005)
30. Amaral, L.A., Hessel, F.P., Bezerra, E.A., Corrêa, J.C., Longhi, O.B., Dias, T.F.O.: eCloudRFID-A mobile software framework architecture for pervasive RFID-based applications. Journal of Networks and Computer Applications 34, 972–979 (2011)
31. Welbourne, E., Battle, L., Garret, C., Gould, K., Rector, K., Raymer, S., Balanzinska, M., Borriello, G.: Building the internet of things using RFID: The RFID ecosystem experience. IEEE Internet Computing 19, 48–55 (2009)
32. Xiaowei, Z., Mukhopadhyay, S.K., Kurata, H.: A review of RFID technology and its managerial applications iin different industries. Journal of Engineering and Technology Management 29, 152–167 (2012)
33. Yi-Wei, M.: Mobile RFID with IPv6 for phone services. In: 13th IEEE International Symposium on Consumer Electronics (ISCE 2009), pp. 169–170 (2009)
34. http://www.abiresearch.com/research/1003525-Near+Field+Communications+NFC
35. http://www.nfcworld.com/2010/06/02/33802/ims-forecasts-785-million-nfc-chips-to-ship-in-2015/
36. http://www.device-solutions.com/downloads/NFC%20Enabling%20Technology%20Final.pdf
37. http://www.rfidjournal.com/article/view/8793
38. Boukerche, A., Turgut, B., Aydin, N., Ahmad, M.Z., Bölöni, L., Turgut, D.: Routing protocols in ad hoc networks: A survey. Computer Networks 55, 3032–3088
39. Hanzo II, L., Tafazolli, R.: A survey of QoS routing solutions for mobile ad hoc networks. IEEE Communications Tutorials & Surveys 9, 50–70 (2007)

40. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Networks 3, 325–349 (2005)
41. Tian, Y., Hou, R.: An improved AOMDV routing protocol for internet of things. In: International Conference on Computational Intelligence and Software Engineering (CiSE), pp. 1–4 (2010)
42. Zheng, J., Liu, Y., Zhu, C., Wong, M., Lee, M.: IEEE 802.15.5 WPAN Mesh Networks. IEEE P802.15-05-0260-00-0005 (2005)
43. Vasseur, J.P., Agarwal, N., Hui, J., Shelby, Z., Bertand, P., Chauvenet, C.: RPI: The IP routing protocol designed for low power and lossy networks. IPSO Alliance (2011)
44. Bocchino, S., Petracca, M., Pagano, P., Ghibaudi, M., Lertora, F.: SPEED routing protocol in 6LoWPAN networks. In: IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1–9 (2011)
45. http://www.freshbusinessthinking.com/business_advice.php?CID=30&AID=10320&PGID=2

Challenges in Efficient Realtime Mobile Sharing

Dennis J.A. Bijwaard, Henk Eertink, and Paul J.M. Havinga

Abstract. Future applications are envisioned to be adaptive to various changes in network, environment and situation. These so-called pervasive applications will be composed from both locally and globally available multimedia resources such as audio and video, web services and context sources. The rapidly increasing pervasiveness in todays networks, i.e. the number of mobile devices, the amount of data they generate and share (near) realtime increases rapidly. In fact this forms the basis for research efforts on the Internet of Things. The increased pervasiveness leads to numerous efficiency and scalability challenges. This chapter will detail the state of the art and binding concepts for efficient realtime sharing and mobility of multimedia and context. Additionally, it lists the associated challenges and their progress in a number of research projects.

In the past decade we have seen a number of technology boosts. Multimedia like audio and video moved from analogue to digital, enabling free audio/video calls over the Internet. Multiple network types have been integrated into mobile devices and bandwidth is gradually increasing. Sensors became wireless and form wireless sensor networks for environmental monitoring, and multiple sensors are added to mobile devices enabling different interaction modalities and situation awareness, The number of mobile devices and their capabilities are gradually increasing, and server clouds have been created that offer remote processing and storage.

Dennis J.A. Bijwaard
Inertia Technology, Enschede, The Netherlands
e-mail: dennis@inertia-technology.com

Henk Eertink
Novay, Enschede, The Netherlands
e-mail: Henk.Eertink@novay.nl

Paul J.M. Havinga
The University of Twente, Enschede, The Netherlands
e-mail: P.J.M.Havinga@utwente.nl

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 115–142.
DOI: 10.1007/978-3-642-34952-2_5

As a result, today's connected networks enable feature-rich applications that adapt according to situational and environmental changes, and the networks and objects that are encountered. Some currently available applications already adapt according to changes in the environment and the situation that the user is in, examples are: route-planners that adapt to traffic conditions along your route, games that react to movement of a handled mobile device, mobile devices that automatically connect to wireless networks that are in reach.

However most of current adaptive applications are dedicated to a single task, and there is only limited sharing of information between applications of different vendors. Furthermore, the performance of applications is often determined by the polling frequency of mobile devices and by the server-side capacity, which is often consumed by distributing the same information to many mobile devices. At the same time, the mobile devices change network and can be temporarily without network. Efficient sharing of multimedia content is nowadays limited to that of dedicated content providers, and the content is not seamlessly continued when changing access network.

Future applications are envisioned to be adaptive to various changes in network, environment and situation. These so-called pervasive applications will be composed from both locally and globally available multimedia resources such as audio and video, web/cloud services and context sources. In fact all devices used in these pervasive applications are elements of the Internet of Things (IoT), all services are part of the Internet of Services (IoS), and all multimedia resources and services are part of the Internet of Media (IoM). Context sources in these pervasive applications can vary from your mobile phone's sensors to dedicated sensor networks deployed in buildings and vehicles, sensor nodes attached to beings and objects, and events generated from devices and applications. Higher level context can be obtained by reasoning based on this sensor information and events. Important prerequisites for interoperability of different services are identity federation for usage of cooperating services, standardisation of discovery and usage interfaces. A prerequisite for context gathering in the IoT is the geographical location where context is gathered. This location often has to be deduced from the environment of a moving entity (e.g. a complete sensor network can move while only one node has Global Positioning System (GPS) support).

Example pervasive applications are sharing your live video with large group of mobile users, automatically switching received video on your mobile to a bigger screen nearby and adjusting nearby light levels for advanced viewing, realtime access to shared context information independent of your current network connection.

Current enabling technologies like Web 2.0, Grids, P2P, and cloud computing may not be sufficient to enable the multitude of mobile users and applications to use and share realtime multimedia content, and context information like sensor network information over the Internet. Also composing pervasive applications from multimedia, web/cloud services and context sources still holds many challenges.

The underlying problem is the rapidly increasing pervasiveness in todays networks, i.e. the number of mobile devices, the amount of data they generate and share (near) realtime increases rapidly. This leads to numerous efficiency and scalability challenges.

This chapter will detail the state of the art and binding concepts with respect to efficient realtime sharing and mobility of multimedia and context, the associated challenges and their progress in a number of research projects. First, mobility and sharing in heterogeneous networks (including sensor networks) (see Section 1) and multimedia (see Section 2 will be described. Next service platforms handling mobility and roaming (see Section 3), and pervasive service platforms (see Section 4) that enable composed services from web, context and multimedia services. This chapter concludes (see Section 5) with the challenges and research progress for efficient realtime mobile sharing.

1 Heterogeneous Networks

In this section we describe different networks that enable users access to the Internet via their devices (See Section 1.1), and Wireless Sensor and Actuator Networkss (WSANs) (see Section 1.2) that gather information from the environment and allow actuation in that environment. User devices and WSAN nodes play an important role in the IoT.

1.1 IP Networks

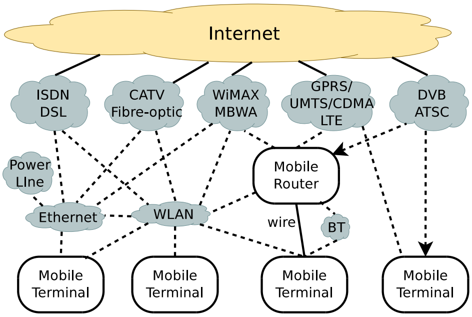
In the beyond-3G environment of today, users have access to an increasing number of different access networks, both wireless and fixed. The combination of fixed and wireless networks enables end-users to be almost always on-line and connected to their preferred network(s).

Beyond 3G-environments include beyond 3G-networks (also called Next Generation Networks (NGN)) as well as next generation terminals and services. Beyond 3G-networks consist of a variety of wireless and wired networks as core and access networks. End-user terminals and service providers are the end-points of these networks. Global IP-connectivity exists between all these networks and all end-to-end communication is IP- based. See Figure 1 for a number of IP connectivity options including unidirectional broadcast networks. Note that multiple access networks can be used simultaneously (multi-homing). With a mobile router, a mobile data connection can be shared with other devices over wireless technologies like Wireless LAN (WLAN) and Bluetooth or wired. When doing this with a mobile terminal this is often called tethering. This creates a potentially moving network, e.g. when used in a train.

In the next subsections we describe transparent mobility with Mobile IP, IP data flows, and broadcast/multicast/unicast.

1.1.1 Mobile Internet Protocol

Mobile IP [35, 28] (MIP) allows transparency of network changes and allows to maintain all TCP/IP connections while changing networks. MIP is mostly beneficial



ATSC	Advanced Television Systems Committee
BT	Bluetooth
CATV	Cable TeleVision
CDMA	Code Devision Multiple Access
DSL	Digital Subscriber Line
DVB	Digital Video Broadcast
GPRS	General Packet Radio Service
ISDN	Integrated Services Digital Network
LTE	Long Term Evolution
MBWA	Mobile Broadband Wireless Access
UMTS	Universal Mobile Telecommunications System
WLAN	Wireless Local Area Network
WiMAX	Worldwide Interoperablity for Microwave Access

Fig. 1 Different IP connectivity options

for connections with longer duration. A lot of tasks on mobile devices (such as web browsing and fetching/sending email) are not troubled so much by network changes since they are done rather quickly and can be easily be repeated when they happen to tail by a network change. Mainly longer sessions like Virtual Private Networks (VPNs), large up/downloads, and multimedia sessions need to be maintained while changing networks.

Mobile IPv4 [36] is the IETF standard for supporting mobility at the network layer in IPv4 networks. The terminal denoted as the Mobile Node (MN) gets a home IP-address assigned to be used for all communications. When the MN is not in its home domain, a so-called Home Agent (HA) forwards (tunnels) traffic to the MN's current location in a foreign network. In the foreign network, the MN obtains a Care-off-Address (CoA) from a Foreign Agent (FA) or a DHCP server, resulting in a FA-CoA (which is the address of the FA itself) or a co-located CoA, respectively. A co-located CoA has the advantage that an FA is not required in every visited network. Each time an MN changes its CoA it must re-register it with its HA in order to receive traffic directed to its home IP-address.

Mobile IPv6 [28][37] addresses a number of the Mobile IPv4 shortcomings such as the triangle routing problem. Route optimisation in Mobile IPv6 circumvents the triangle routing problem by sending binding updates, containing the current CoA of the MN, from the HA to all correspondent nodes.

Extensions have been proposed to Mobile IP to also handle moving networks with Network Mobility (NEMO). Examples of such moving networks are trains and planes that share their connection to a celular network like Universal Mobile Telecommunications System (UMTS) with the people they transport.

1.1.2 IP Data Flows

Communication in heterogeneous networks is a combination of data flows between applications. These data flows can be connection oriented with protocols like Transmission Control Protocol (TCP) and Stream Control Transmission Protocol [46] (SCTP) or connection-less with protocols like User Datagram Protocol (UDP).

These flows can be protected from eavesdropping using security measures, and their quality can be maintained using Quality of Service (QoS) measures.

Security of IP data flows can be done at multiple layers of the TCP/IP model:

- at the network access layer by encrypting the packet payload
- at the Internet layer by using Internet Protocol Security [29] (IPsec)
- at the application layer by using protocols like Secure Socket Layer (SSL) and Transport Layer Security [21] (TLS) for secured bidirectional connections, and Pretty Good Privacy [17] (PGP) for securing individual messages.

In order to provide QoS, packets of separate IP flows can be classified differently (e.g. as best-effort, audio and video), such that they can be treated properly in the network. QoS treatment involves all network layers in every network element in the communication path, as illustrated in Figure 2. End-to-end QoS is determined by the lowest weakest link among all network elements between sender and receiver, and end-to-end QoS can be solved by dividing the problem along network domain boundaries [32], as illustrated in Figure 2.

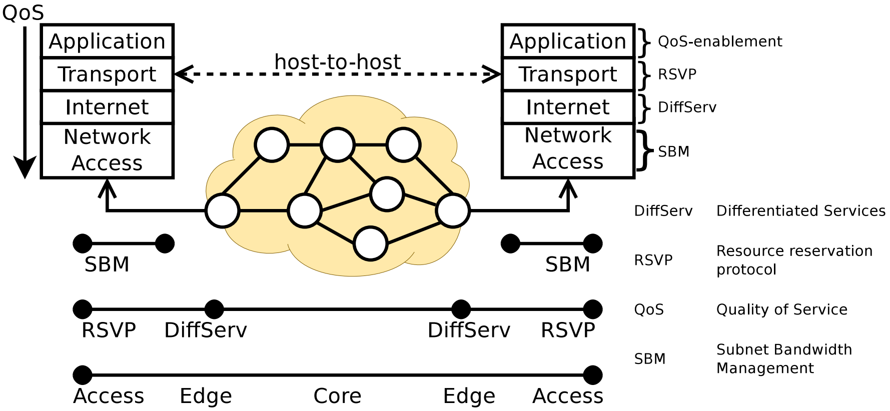


Fig. 2 End-to-end QoS across access, edge and core domains

1.1.3 Digital Broadcast, Multicast, Unicast

The main difference between broadcast, multicast and unicast is that broadcast is destined to everyone that is able to listen. Multicast is for a selection of listeners, and unicast is directed to a specific listener. A distinction can be made between bidirectional broadcast in which the same (wireless) medium can be used to send something back and unidirectional broadcast that is only one way. Unidirectional broadcast can use a return channel on another access medium to send something back.

Traditional broadcast uses (radio) technologies to broadcast content to a large number of users, such as analog audio channels and Television (TV) via the air or via cable, and the last decade digital broadcast gradually became the new standard with mainly Digital Video Broadcast (DVB) and to a lesser extend Advanced Television Systems Committee (ATSC).

In fixed telephony, any of various Digital Subscriber Line technologies (xDSL) is used for multicasting television to the end-users and offering interactivity with Internet Protocol television (IPTV). Broadcasting all channels is not really an option in xDSL since the last hop to the user is the dedicated twisted pair phone line which currently only supports high data rates over small distances.

In the mobile telephony standards UMTS and Code Division Multiple Access (CDMA), Multimedia Broadcast Multicast Service (MBMS) offers multicast and broadcast on handsets and via data cards (e.g. for laptop). MBMS is an enhancement feature of the UMTS architecture aiming at providing the capability for Broadcast and Multicast Services in the network (under Release 6).

DVB is availabe in a number of types, including DVB-Satelite (DVB-S), DVB-Cable (DVB-C), DVB-Terrestrial (DVB-T), and Digital Video Broadcast - Handheld (DVB-H). All DVB data and digital data in ATSC is transmitted in Moving Picture Experts Group (MPEG) transport streams, which enables transmission and storage of audio, video, and data.

There are basically two types of multicast over IP, namely Source Specific Multicast [12] (SSM) and Any Source Multicast [19] (ASM). In ASM the user expresses its interest in a specific multicast group, in SSM, the user expresses interest in a combination of a specific source and multicast group. In both cases the routers between the source and destination need to make sure that the users that joined the multicast group get the associated IP streams efficiently (without unnecessary duplication).

1.2 Wireless Sensor and Actuator Networks

A WSAN typically consists of a large number of low-power sensor and actuator nodes. These nodes are equipped with a wireless transceiver, a small microcontroller, a power source and multi-type sensors such as temperature, humidity, light, heat, pressure, sound, motion, etc. Additionally, the nodes can be equipped with actuators such as Light Emitting Diodes (LEDs), switches, and even motors. WSAN nodes are some of the smaller devices in the IoT that collectively generate context information that can enhance pervasive applications. When these WSANs also have processing capabilities, they are also referred to as Pervasive Systems, i.e. systems containing a large number of collaborating tiny sensing, actuating, routing, and processing devices.

WSANs are commercially available in various forms, shapes, sizes, and functionality running various operating systems (e.g. TinyOS [16] or AmbientRT [26]). Interaction between sensor nodes and applications has not yet been standardized.

Applications involving WSANs are very diverse and involve one or a combination of various types of sensor networks. We can identify at least six types of wireless sensor networks, namely (based on [31]):

- **Environmental Sensor Network (ESN):** These are the very first type of wireless sensor networks. Traditionally, ESNs were solely deployed for monitoring and data collection purposes. ESNs are often large scale, static, non-dense, and are deployed in harsh and unattended environments. Energy efficiency, long network life-time and security have always been the major concerns of ESNs.
- **Body Sensor Network (BSN):** BSNs are sensor networks consisting of few wireless sensor nodes on or around a living being's body connected to a more powerful device such as a smart phone. Monitoring of vital signs, tracking, and data collection have been the main objectives of these sensor networks. Interaction with sensor-enabled objects [15], such as a dumbbell or ball, is an interesting upcoming usage area. BSNs are small scale, use different types of sensors and are usually limited to single-hop wireless communication. Since personal information can be collected by these networks, both security and privacy are major concerns.
- **Structure Sensor Network (SSN):** SSNs consist of medium to large numbers of wireless nodes usually attached to or in buildings (e.g., office), structures (e.g., bridges), infrastructure (e.g., rails) or deployed in specific venues (industrial sites). Wireless nodes can also be attached to objects moving inside the structure and between structures. SSNs usually extend their wireless coverage with multiple hops of wireless communication and often use a variety of sensors.
- **Transport Sensor Network (TSN):** Transportation means such as cars, trucks, and trains, have a number of sensors. Over the past few years, many efforts have been directed towards wireless communication and networking between transportation vehicles (e.g. vehicle to vehicle communication via IEEE 802.11p). Each individual vehicle can be seen as a sensor node, which locally observes its own state while it also monitors its surroundings.
- **Vehicle Sensor Network (VSN):** The sensor data from within a moving vehicle (e.g. a car, boat, train, plane) can also be transferred wirelessly (e.g. via General Packet Radio Service (GPRS)) to a central server, and be monitored remotely and/or merged with data from other sensor networks. In warehouse logistics, VSNs are often used together with SSNs, e.g. when monitored goods are transported in a truck from one warehouse to the other.
- **Participatory Sensor Network (PSN):** Mobile phones are becoming more and more equipped with sensors (e.g., GPS, accelerometer, gyroscope, camera) and different types of connectivity mediums (Bluetooth, wifi, Global System for Mobile Communication (GSM), etc.). This combination makes the mobile phone and in fact people carrying them a valuable source of collecting and transmitting information. Information collected by people through their mobile phones can range from personal health conditions and their trajectory to environmental conditions and pictures of the area in which they move around.

Mobility is typically covered within the WSAN, i.e. nodes within the WSAN can move around and use alternative nodes to stay connected. Mobility of nodes across WSANs and mobility of Internet-connected WSANs that are potentially used by multiple applications are still research topics.

The following application areas are considered [31], where WSANs are mobile and are potentially used by multiple applications:

- **Cool Chain Logistics:** In the cool chain market, it is important to optimise the quality of perishable products by ensuring optimal storage and transport conditions. In addition, assets can be tracked when they enter or leave certain areas.
- **Environmental/Habitat Monitoring:** Monitoring is done in the environment or the habitat of living beings, usually for extended periods where user-intervention is either expensive or disturbing. Data mules are sometimes used to collect the sensor information when no wireless coverage is available. In habitat monitoring also the animals themselves can wear a sensor node.
- **Surveillance:** Building, vehicle and infrastructure monitoring to detect forcefully opened or unlocked doors/windows, theft and damage.
- **Smart Spaces:** Smart spaces adapt to the needs of the users that enter and leave. They typically contain sensors and actuators that can be monitored and controlled by applications running in the environment and on user devices.
- **Remote eHealth:** In remote eHealth, sensor networks consist of few wireless sensor nodes on or around a living being's body. Typically, these nodes are integrated with a smart phone or a stationary device at home. Monitoring vital signs, and tracking are the main objectives of these sensor networks. Analysis is often done offline but increasingly becomes real-time.

Table 1 lists which WSAN types are typically used in each application area, and what items are mobile.

Table 1 Typical associations in specific application areas

Area /association	Cool chain logistics	Environment monitoring	Surveillance	Smart spaces	Remote eHealth
Mobile entity	truck, node	data mule, node	vehicles	user-device, object	user-device
Domains	depot, warehouse	geographic area	building, infrastructure	place	clinic
WSANs	areas, trucks	sub-areas	vehicles, areas, different types	different types	patients
Nodes	roll containers	sensor node	door, window	sensor nodes	sensors, objects
WSAN types	SSN, VSN	ESN	SSN, VSN	BSN, SSN, PSN	BSN
Apps	views, triggers	views, triggers	views, triggers	experiences	views, feedback

2 Multimedia Sessions

Multimedia sessions are sessions that contain one or more multimedia streams. In this Chapter we mainly focus on audio and video streams. Examples of multimedia sessions are Voice over IP (VoIP), audio/video teleconferencing, Video on Demand (VOD) and IPTV. Multimedia session enablers are part of the IoM.

This section first describes protocols for multimedia session control, then mobility for multimedia sessions and then compares multimedia session mobility with mobileIP.

2.1 Session Control

For controlling realtime multimedia sessions over the Internet between two or more parties, a number of standards are available, namely:

- The Real Time Streaming Protocol [44] (RTSP) supports video-like control over a multimedia session with a streaming server. It can for instance be used to establish, play, fast-forward, pause and stop a multimedia session containing multiple media flows;
- Revision 5 of HTML (HTML5) which is still under development supports playing audio and video files and is expected to support realtime multimedia playing in a web browser using RTSP.
- The Jingle [24] protocol extension to Extensible Messaging and Presence Protocol [42] (XMPP) enables signalling via an XMPP server for multimedia session setup;
- H.323 that uses telephony-style signalling from the International Telecommunications Union Telecommunications Sector (ITU-T);
- Session Initiation Protocol [11, 39] (SIP) using HyperText Transport Protocol (HTTP)-style signalling from the Internet Engineering Task Force (IETF);

Apart from those, closed approaches are available such as Skype and the flash player.

RTSP and HTML5 are mainly used for controlling unidirectional multimedia either from or to a streaming server and are not further considered. Jingle, H.323 and SIP do support setting up a multimedia session with multiple multimedia streams in any direction. Jingle does not support session mobility yet. There is one extension for session transfer [50] that has the deferred state. Jingle is designed to interwork with SIP. Because of the current lack of session mobility, this protocol is not further considered.

H.323 [27] is a standard published by the ITU-T for audio, video and data communication across IP networks. The H.323 Recommendation can be applied to voice-only handsets and full multimedia video-conferencing endpoints, and others. H.323 is part of the H.32X series for enabling video-conferencing across a range of networks including Integrated Services Digital Network (ISDN), Public Switched Telephone Network (PSTN) and IP networks.

H.323 does only provide seamless mobility while roaming when the network point of attachment does not change during handover (see recommendation H.510 from the ITU-T, e.g. when a mobility mechanism like MIP is in effect, or when all communication is tunnelled to the home provider network). H.323 is not further considered in the remainder of this Chapter.

SIP, as described in [43] and [41], is a signalling protocol used for establishing, maintaining, and terminating multimedia sessions and providing presence information in an IP network. Traditionally, resource discovery in SIP is done in a centralized manner, i.e. each domain has a local resource directory where all identities and their preferences are stored. SIP is adopted by the IP Multimedia Subsystem (IMS) of the 3rd Generation Partnership Project (3GPP) [1]. Peer to Peer (P2P) SIP, offers a distributed mechanism for resource discovery which can reduce (or even eliminate) the need for centralized servers. In the remainder of this chapter only traditional SIP is considered unless specifically stated otherwise.

SIP can, in addition, provide user mobility functionality because the identification of users with SIP is independent of underlying IP addresses. Wedlund and Schulzrinne in [49] proposed to use mobility support in SIP to support real-time communication. Most current SIP user agents on mobile terminal do not support these methods.

2.2 *Multimedia Session Mobility*

SIP has its own mechanisms for mobility management [49] for SIP-based applications as well as functionality for session adaptation.

Application layer mobility solutions, for example based on SIP, can either replace or complement network-layer mobility [45].

No single approach to IP mobility applies across heterogeneous applications in beyond 3G environments [33]. To meet the requirements of applications and deal with harsh networking environments multi-layered mobility management solutions and architecture are proposed, see for example [22] and [38].

2.3 *Multimedia Session Mobility versus MobileIP*

There have been a number of studies comparing SIP-based and MIP-based mobility management. The comparisons of the performance of the two protocols in [49] and [9] demonstrate that, in general, application-layer mobility management protocols, such as SIP, perform worse than lower-layer protocols in terms of hand-off delay, signalling overhead, and transparency. However, when suitability for deployment in next-generation networks is considered, it appears that SIP is a better mobility management solution for multimedia sessions, because it obviates the need for protocol stack and infrastructure changes [9]. A number of studies indicate that the suitability of a mobility management solution depends primarily on the type of application for which it is being considered. For long-lived TCP connections (such as FTP) and most standard Internet applications (such as Web browsing and chat), MIP offers a generic solution for roaming that seems to work well. However, for real-time applications, SIP is recommended [48, 49], because real-time applications (e.g., multimedia applications) have strict timing requirements that are not taken into account by MIP because it is a network-layer protocol. To optimize roaming behaviour, applications should be able to influence or even control the mobility management process, as they can when SIP is used as the mobility management solution. An additional benefit of using SIP for application-layer mobility management is that it allows applications to adapt their service behavior, based on the mobility management strategy selected, to provide the best possible end user experience.

3 Federated Service Platforms

Service platforms (see Figure 3) enable access to service providers to devices that are connected via heterogeneous networks. Federation between service platforms realises a service control layer. This layer enables third-party service providers to offer their services to roaming end-users, while being shielded from network-specific details. In addition, end-users with a Service Platform subscription can use the services to which they are subscribed while switching access networks (including foreign ones). The Service Platform adds value for functionality such as mobility management, session control, authentication, user profiles, and user localization.

We first describe the functionalities of service platforms, then further detail mobility management.

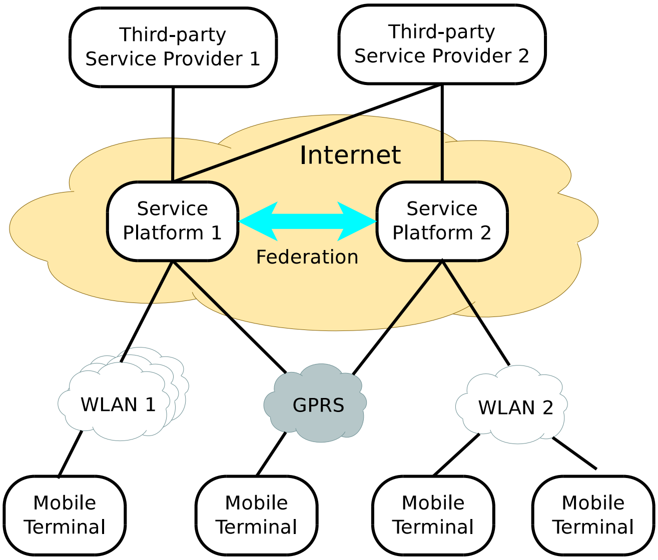


Fig. 3 Federated service platforms enabling services to mobile terminals

3.1 Functionalities

There are a number of different functionalities that a Service Platform can offer, among others:

- Bridging legacy systems: E.g. a multimedia gateway can be used to bridge telephony between GSM/PSTN/ISDN networks and VoIP.

- Providing identity across federated service platforms: a user has a home service platform and can use its identity to use foreign access networks and services provided by other service platforms. This federated identity is one of the prerequisites for the IoS.
- Multimedia and messaging: E.g. SIP application server, to handle voice, video, and messaging applications. This functionality is an enabler for the IoM.
- Charging: real-time and offline charging for services. Example charging types are time-based, volume-based, event-based.
- 3rd Party service interfaces: Example standards are the OSA/Parlay and their RESTful successors [8] for Telecom-based service platforms.
- Seamless use of different Access Networks: i.e. providing mobility management, see Section 3.2.

3.2 Mobility Management

To facilitate seamless continuation of services across access networks, users should be able to roam seamlessly from one access network to another and/or attach to multiple access networks simultaneously (multi-homing). Mobility management, which is the technical prerequisite for roaming behavior and service access, involves controlling the network(s) to which the user's terminal is connected and which service runs through which access network. I.e. mobility management discovers new access networks, and controls the handover between these networks [10]. It is also responsible for roaming services (e.g. continuous access to SMS and IP services). The services that can be supported on or across access networks depend on the characteristics (e.g., the bandwidth restrictions) of these networks; certain services may not be supported on certain networks. Therefore, it may be necessary to adapt ongoing service sessions to changes in the network environment. A typical example of such an adaptation is dropping video from an audio-video session for a low-bandwidth access network.

Mobility management plays a key role in dealing with user and terminal mobility in beyond 3G-environments. Following [6] and [7], mobility management can be defined as a functional component that firstly keeps track of the IP-addresses of mobile end-users, and secondly modifies the IP routes of the ongoing sessions of mobile end-users[1]. The mobile end-users' IP-addresses can be tracked per session. This Mobility management function enables other end-users to initiate new sessions towards the mobile end-user. Similarly, modification of the IP routes of ongoing sessions can be done collectively (for all sessions) or individually (per each session). Modification of active sessions is subject to the requirements of the sessions involved; examples of such requirements are minimal bandwidth and cost.

A mobility management system in the beyond 3G-environments described above has the following characteristics:

[1] A session can be an instantiation of a service that is established between two or more endpoints (i.e., users and/or machines). A more elaborate service concept is described in 4.1 and session concept is described in 4.2.

- Mobility management concerns[2] both user mobility and terminal mobility aspects. Therefore, the end-user (and not just her/his terminal) is the one whose mobility is tracked and handled by the mobility management.
- An end-user is likely to be associated with multiple IP-addresses corresponding to the active network interfaces of her/his terminal(s).
- Due to diverse requirements of heterogeneous applications in beyond 3G-environments, no single approach to IP mobility applies across these applications [33]. Therefore, the mobility management should provide multiple IP mobility solutions at different layers of the OSI model to handle mobility issues for services, individually or collectively.
- This asks for a multi-layered mobility management approach (i.e. mobility at different layers of OSI model) where the scope of the mobility management spreads from each individual service (and its sessions) to an aggregation of all services (and their sessions), associated with an end user. In other words:
  - The IP address to be tracked by the mobility management is the routable IP-address of the terminal interface, to which the end-user is attached for initiating a session of a particular service or any subset of services (this subset can include all her/his services).
  - The IP route to be modified by the mobility management corresponds to the terminal interface, via which the end-user is involved in an ongoing session or in any subset of ongoing sessions (this subset can include all her/his sessions).
- At any given time, the IP-address of an ongoing session of a service can differ from that for initiating a new session of the same service.

4 Pervasive Service Platforms

A pervasive service platform (see Daidalos [4]) offers user experience in addition to the Service Platform. This user experience can be composed from existing services and be personalized and situation aware, by utilizing sensor information, context, profiles, and history of the user and the environment. Even when the user is not connected itself, the pervasive service platform and services can act on behalf of the user. The context-aware and personalized service composition is an enabler for the IoS, and context-aware multimedia services are an enabler for the IoM.

Pervasive Service Platforms are a distributed form of pervasive systems that provide a home base for the users, and give them a digital identity at that base. Federation of platforms allows the user to communicate with users at other bases and use services provided at other bases. The federation of pervasive service platforms form an enabling middleware for the IoT.

Other forms of pervasive systems can be organized as peering components (see e.g. Hydra [23]) that can discovered and hooked-up dynamically, for instance when they get close to one another.

[2] A full solution involves the cooperation with other system functions like AAA, personalization, session control, etc.

In this section we describe the concepts for services, sessions, mobility and sharing in pervasive service platforms (see Daidalos deliverable DII-124 [5]).

4.1 Service Concept

A *service* is defined as a (potentially distributed) software application that provides certain functionality accessible via well-defined communication protocols. The type of service is defined by the offered functionality and the supported access protocols. Service sessions are running implementations of the service's functionality and protocols. According to this definition the type of a service is defined by the communication protocols it supports and the functionalities it uses and offers. This definition allows us to include a wide field of services including data services (e.g. a currency translator or email) and usage/configuration of hardware devices (e.g. a display or a printer). A service session is a concrete implementation of a service type that is actually running. Services often follow a traditional publish/discover/subscribe paradigm, meaning they are registered on a server, can be discovered by querying this server, and once discovered a service can be accessed directly.

Table 2 Pervasive Service requirements

Characteristic	Requirement	Short description
Discoverable	Required	A pervasive service has to expose its functionality, the supported protocols and its attributes in a standardised way, independent from the particular service discovery protocol. Nevertheless, it has to support at least one (e.g. SLP).
Composable	Required	A pervasive service needs to be able to cooperate with other services.
Context-aware	Optional	A service may be context aware, i.e. adapt according to for instance situational, network or environmental changes.
Personalisable	Optional	A pervasive service may be aware of the user's personal preferences, i.e. it may have parameters that can be personalised.
Private and Secure	Optional	A pervasive service may specify privacy and security requirements when accessing sensitive user-related data.

A *pervasive service* is a service that exposes its functionality and attributes in a standardized way, and is made available via specified service discovery protocols. The service can be integrated into a composite service. It may be security and privacy aware, context aware and allow for personalisation. Table 2 summarises the six requirements for a pervasive service.

The whole concept of the pervasive services is their adequacy as building blocks for more complex services, denoted as composite service: A set of cooperating pervasive services. A composite service may also be a pervasive service (recursive definition). A running composite service session is called a (composite service) session. Since service sessions that are being part of a composite service session, it may come and go frequently and necessary context information may change often, and therefore re-composition may need to be carried out regularly. During recomposition, service sessions might be added, reconfigured, removed or replaced by alternative ones. At a certain point in time the composite service session will be terminated, i.e. the composite service is stopped. During this process the individual services are disconnected and released. A composite service may also be called an application since the composite service provides the functionality to the end user. In order to create a composite service, knowledge is needed about how a composite service shall be created. This can be done both by the network provider, or by third parties.

4.2 Session Concept

When discussing data connections and streams it is necessary to describe the relation between the data packets, terminals, network nodes and services. This relation is usually known as a session, and may not always be easily identified (e.g. the set of packets under scope of a SIP application may not be identifiable by the traditional double set of IP addresses and ports). Another characteristic of a session is that it defines the relationship between a set of network nodes. For instance, a SIP session will involve a number of network nodes (SIP clients, SIP proxy) that are involved in the exchange of packets. The scope of what a session is varies with the aspect we are tackling. The following types of sessions are identified:

- **Network Access Session:** Between a device (mobile terminal) and a wired or wireless network. Especially for wireless networks, this usually involves authentication to the network.
- **Network Identity Session:** Includes all Network Access Sessions that use the same identity (or credentials) for authentication. This means e.g. having multiple virtual or physical interfaces on a mobile terminal authenticated using the same credentials.
- **Transport Session:** A transport session connects and/or exchanges data to and/or from a node in the network. Multiple transport sessions can exist within the same network access session, typical examples of transport sessions are TCP connections and UDP streams.
- **Application Session:** Contains (zero or more) transport sessions. Can exchange application-specific packets among distributed application parts. An important subtype of the application session is the multimedia session.
- **Pervasive Session:** A session that is directly mapped onto user goals and intentions. Can be context-aware and personalizable. Will control overall coordination of multiple application sessions that might interact with each other based on user context.

The following paragraphs further detail these session types.

Network Access Session

The network access session starts with authentication of the user identity and checking authorisation for its access from a given mobile terminal on a given network interface to a specific access network. When authentication is not necessary, the network access session starts by connecting to the network. The network access session ends when the network access is terminated on that network interface, which could be the case when a re-authentication is necessary in another network (e.g. for inter-domain mobility) or when the user logs off the mobile terminal etc. All other sessions have to be supported on top of these network access session(s).

In relation to the access technology, a network access session is always bidirectional, even when bidirectional technologies are being emulated using Unidirectional Link Routing (UDLR). Nevertheless, we can be have mostly unidirectional (e.g. authenticated + authorised broadcast access) or mostly bidirectional (e.g. wifi, UMTS, broadcast access + return channel, etc.) access. This does not preclude for a user with a DVB-enabled mobile terminal to access free information on the DVB broadcast, or, after registration, to keep on receiving protected content for a given time without any uplink (until registration times-out).

A network access session can have QoS guarantees as a whole, or for its contained transport sessions. Also both the whole session and the sub-sessions can have associated costs. The QoS guarantees and costs can differ when the network access session is handed over to another network, or to another mobile terminal. During a network access session, the Home Address (retrieved based on selected identity) of the interface will stay the same, and only the CoA will change after a handover.

Transport Session

Within a network access session, several transport sessions can exist, the most important ones are TCP, UDP and SCTP. TCP is connection oriented, is bidirectional and provides ordered reliable transport. TCP sessions have a clear start and end. UDP provides messaging over IP, and these messages can be combined into streams between the same sender and receiver(s). The end of a UDP session is much harder to determine and an UDP session is usually unidirectional (or multi-directional in case of multicast traffic). SCTP is message-oriented and can also ensure ordered and reliable transport like TCP, on top of that it supports multiple data streams in parallel within the same SCTP session and can support transparent failover in multihoming scenarios. SCTP sessions have a clear start and end like TCP.

While authentication and authorisation are never done on the granularity of a transport session, QoS control and mobility management are typically referring to transport sessions. In case of a handover between different domains that requires re-authentication, a transport session will typically continue, while the related network access session changes.

Network Identity Session

A network identity session contains all concurrent (i.e. overlapping in time) Network Access Sessions that use the same identity for authenticating to the network(s). The same identity can be used for connecting to different physical interfaces using different technologies, or to different logical interfaces on the same physical interface. As long as the Network Access Sessions use the same identity they are considered to be part of the same network identity Session.

Application Session

Over these access sessions, applications are running their own sessions. One example of an application session is a multimedia session; other examples are broadcast services, context services, web services, and simple applications like telnet and ftp. The simple applications and web services are more traditional sessions, and easier to identify. The complexity of application sessions is reflected in complex services, even if not multimedia. Other non-multimedia sessions may cover services such as lookups for a restaurant, or cinema, access to traffic (jam) data, or download of audio/video content for offline consumption (e.g. buying songs on iTunes). Some of these services may actually relate multiple applications, and involve changes of many connections and/or connection end-points. For those application sets an application session is comprised of all data transactions which take place until a service transaction (e.g. buy and download an mp3) is finished.

Any kind of control traffic (like multimedia session setup, RTCP control messages, TCP acknowledgements) related to an application is part of the application session (typically consisting of several transport sessions), and that applications sessions may as a whole be subject to mobility and QoS management. The level at which this can be globally handled depends on the specific session characteristics. The multimedia session is further detailed below.

Multimedia Session

A multimedia session (see also Section 2) can be established between two or more endpoints (users or service). The session usually starts after an invitation of a participant is accepted by another participant. A multimedia session can contain a number of multimedia streams and connections between the endpoints (usually audio, video and/or instant messaging), and session-specific messages (signalling) can be exchanged between the session participants. During the session a number of things can change:

- Multimedia streams/connections can be dropped, added or moved to other endpoints (partial session mobility)
- Quality of multimedia streams/connections can be changed
- Endpoints of the session signalling can change network location
- Endpoints of the multimedia streams/connections can change network location (e.g. handover or load-balancing when multihomed)
- Endpoints may join or leave the session
- Session may be transferred to another endpoint.

A multimedia session usually contains a control session on the control plane that defines and controls media session endpoints on the data plane. Figure 4 shows how the control session on the control plane and the media session endpoints at the data plane are related to each other for a typical audio/video session between a mobile node and a correspondent node.

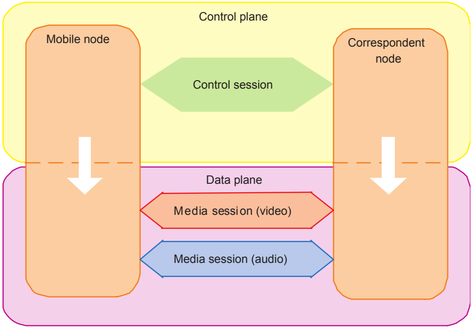


Fig. 4 Multimedia session with control, audio and video sessions

Pervasive Session

A pervasive session, in its simplest form, can be considered a collection of application sessions, both multimedia and web service (or other) sessions. A pervasive session will run a pervasive service, where the different parts of the service can be engaged in different application sessions (e.g. in a multimedia session and/or FTP session). Note that a Pervasive Session is contained inside an identity session. A pervasive session converges application sessions for a specific user experience, where the convergence is guided by the application logic needed to satisfy the user needs. So, while most connections and streams in a pervasive session could be setup by application sessions like the multimedia session, the pervasive session also contains the logic for context-aware starting and stopping of these sessions, adaptation of itself and its parts.

As an example consider a unified conferencing (UC) pervasive service (running in a network node) that is in charge of supporting virtual meetings between three persons in any possible way (depending on the communication technology that is available at any given context for the two people). If initially person A wants to use UC to meet person B there are different scenarios that can be supported by UC:

- Both A and B have mobile phones available to them and can be engaged in a phone conversation. In this case UC could set up a SIP session between A and B.
- While engaged in the SIP-based voice session, A and B might want to share a document. UC might initiate an FTP session or issue a SIP instant message to allow A send a file to B, and later to set up a data conferencing session where A and B can collaboratively browse through the document.
- If A or B's context changes in such a way that one of them gets access to a video camera (e.g. after having moved to another room while in the conference) UC might decide to add a video session to allow A or B see the other party.
- A third person C comes into the conference. C has access to only a chat program. UC might choose to add a speech-to-text conversion session so that C can participate in the conference without disturbing the conversation flow.

- UC might detect that C cannot receive documents on FTP and does not have any document sharing tool available. UC might decide in this case to print the document under discussion to the printer that is located close to C.

The above example shows many characteristics of a pervasive session. UC comprises a pervasive session that implements a multiparty conference among three people. This conference session can be regarded as an overlay application session that is continuously initiating other application sessions depending on the needs of the conference. For all these sub-sessions, the pervasive conference session keeps track of states of the different nodes, is informed about new nodes (e.g. a printer) becoming available, etc. An extended definition of a pervasive session might include application-initiated management of network access and transport sessions. It is easy to see the usefulness of such a concept. For instance, UC above could be extended to make active use of interface selection in order to set up and tear down network access sessions. As long as none of the participants in a conference have access to multimedia tools, UC might choose to use a low-bandwidth low-QoS network connection. Once users get access to multimedia, UC might choose to initiate network access and transport sessions to use better quality network available.

To summarize, a pervasive session has the following properties:

- It is a session that is often long-lived. I.e. it might live in the background and respond to stimuli from its surroundings (e.g. start an application session when there is a context change).
- It is heterogeneous, and will contain different types of sub-sessions. In its simplest form, these sub-sessions will be all application sessions. In its extended form, sub-sessions might also include network access and transport sessions. The pervasive session is the overlay session that manages the sub-sessions.
- It might include resources from different administrative domains. This means that setting up and tearing down sub-sessions might involve federation, authentication, etc.
- It will have several states. It might be running (actively using resources), suspended (not reacting to any stimuli), waiting (in the background, reacting to stimuli) etc.

4.2.1 Session Relationship Example

Figure 5 gives an example of intertwined session relationships. It shows that a *network access* session can exist for different access technologies and that all *network access* sessions for one identity are within the same *network identity* session. It also shows that an *application* session can potentially contain multiple *transport* sessions, e.g. a multimedia session (e.g. S3) can contain both a multicasted UDP stream and a TCP connection for signalling via different access technologies (DVB-X + GPRS), a broadcast session (e.g. S1) can contain multiple multicasted UDP streams, and a WebService session (e.g. S5) can contain one or more TCP connections. A context service (e.g. S10) can get context from *application* sessions, sensors on a user device and sensor networks connected via different identities

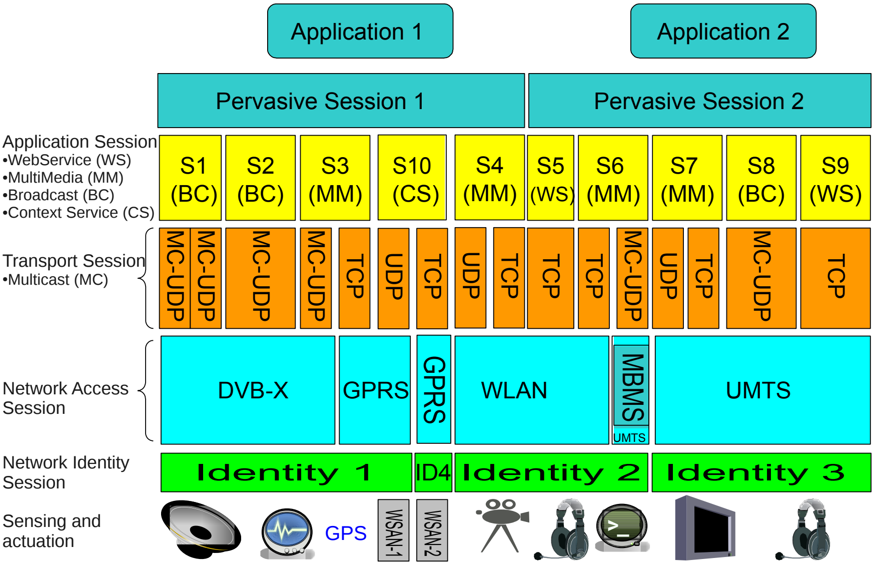


Fig. 5 Example Session relationships

(e.g. identity 1 and 2 over GPRS) and offer this context information or an inferenced version thereof for usage in all constituents of a pervasive session.

At a higher level, Application 1, could start Pervasive Session 1 which contains multiple Application sessions, namely S1, S2, S3, S4 and S10. Application 2 could start Pervasive Session 2, containing Application sessions S5, S6, S7, S8 and S9.

Regarding the relationships above, a set of information is needed in order to keep the overall complex view consistent at runtime. Some of the needed shared information is listed here:

- Identity-related information: The identity used to form a *network access* session will be used for associated *transport* sessions. Utilizing the service platform as an identity provider, this (network) identity could also be re-used by the *applications* and *pervasive* sessions within the *network access* session. However, the identity for those *application* sessions does not need to be shared.
- Preference outcomes: Preferences for using a Network Access Session might depend on the Application Session running on top of it. This information needs to be communicated.
- Context information: Context information might affect how lower-level sessions are configured. Moreover, access to different *network access* sessions might guarantee or deny access to different sets of context sources.

4.3 Mobility

A distinction is made between the following types of mobility:

- **user mobility**, a user can access the network from multiple devices, i.e. the user actually is able to connect and act in a seamless way from all mobile terminals.

- **device mobility**, a device can change its attachment point to the network, i.e. it handles mobility of network access sessions between different access networks. When re-authentication is necessary, the change from one network access session to another would also be considered terminal mobility (but may not be considered seamless and may break running sessions).
- **interface mobility**, a session can be moved from one interface to another in the same device.
- **service mobility**, the provider of a service can be moved during the provisioning of that service.
- **session mobility**, a session or its parts can be transferred between devices.
- **WSAN node mobility**, a node moves within the WSAN or between WSANs.
- **WSAN mobility**, the WSAN may move and therefore change its network point of attachment (device mobility) or change to another network interface (interface mobility).

Different abstraction layers can be considered, both on the network side and the terminal side to abstract technology specific issues, enabling both local and remote communication for enhanced handover procedures. Handover can be initiated either by the mobile node (mobile initiated handover) or by the network (network initiated handover). More advanced concepts such as network aided mobile initiated handover can also be considered.

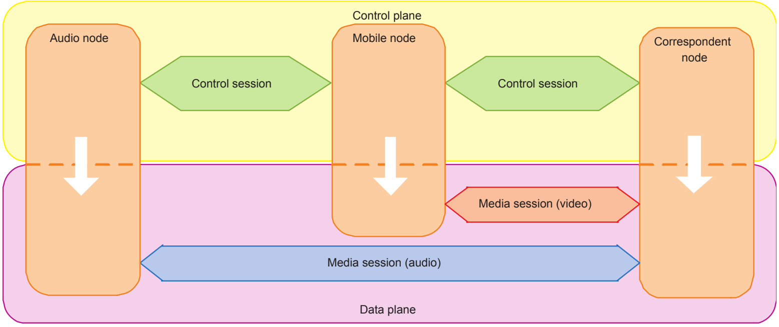


Fig. 6 Audio stream endpoint moved from Mobile Node to Audio Node

Session mobility can be related to mobility of: network access sessions, transport sessions, application sessions and pervasive sessions with associated transport sessions (streams and network connections). Since application and pervasive sessions can be composed of sub-sessions and services that use multiple network access sessions, session mobility can have many aspects, and may cover several of the above mentioned mobility types:

- **Network Access Session Mobility:** the network access session is changed from one network interface to another, or is moved to another device. This would require support for multihoming from the network access provider.

- Transport session mobility: a transport session is moved from one interface to the other or between devices.
- **Partial Session Mobility:** either signalling/control or contained connections/ streams move
  - Multihoming: part of a session moves from one to another network interface
  - Multi-device: part of a session moves from one device to another device (see Figure 6)
- **Full Session Mobility:** the whole session moves
  - Multihoming: The whole session, including signalling, is moved from one interface to another.
  - Multi-device: The whole session, including signalling, moves to another device.
- **Service Session Mobility**
  - Multihomed service session: Part of a composed service session moves from one to another network interface of a device (could be service session on 3rd-party server or on terminal).
  - Part of a composed service moves from one device to another device (could be 3rd-party service or service on terminal)
  - When moving to another domain a candidate service can be instantiated in that domain when the user, personalisation, or context indicates a that service to be similar enough. Such a replacement of service instantiations is also called re-composition.

4.4 Sharing of Content and Context

Content and context (including sensed information) can be shared from (mobile) sources to multiple (mobile) destinations. Mobility here means that sources, destinations and intermediate nodes can move and be temporarily unavailable. Movements of source and destination may also happen simultaneously.

Realtime content and context have a notion of freshness and priority. In a lot of situations, older data is no longer relevant after a temporal outage or limited available bandwidth, and can be discarded, such as with video broadcast. In other cases, such as cool-chain logistics, the history of context data needs to be recorded but can arrive later.

When the number of destinations increases, unicasting to all destinations will consume more bandwidth and processing power. To overcome this bottleneck, one data stream can be sent towards a group of destinations and be divided there (e.g. multicast). Also for checking who is allowed to get the data, the source may not be able to handle all requests when the number of destinations increases.

The sharing can also be influenced by the destinations, not all destination may require the same rate or selection of information. Therefore, remote configuration is required, in a controlled manner. The configuration of one destination, should not affect the experience of another destination. For instance when a destination requires a context update every 5 minutes, the others can still get it at the default 15 minutes and the source could sent it more frequently towards the first destination. Something similar holds for actuation of the source, destinations can potentially sent conflicting actuation commands, so control is required to determine who has authority and priority to make these changes.

5 Research Challenges and Progress

This section describes research challenges and progress in a number of research projects for realtime mobile sharing of multimedia and WSANs.

5.1 Mobility and Sharing in Heterogeneous Networks

For network mobility we distinguish changes in network attachment of devices[1] (such as mobile phones) and mobile networks (such as a WLAN in the train). A user would typically want to use the network or combination of networks that offers the best cost, bandwidth and latency properties. And when networks are no longer in reach, he/she would rather continue the applications seamlessly than restarting them manually. Moreover, when multiple users are using the same wireless network the user would not like his video stream to be interrupted by less time-critical traffic such as file downloads. The challenge is therefore to offer seamless mobility across heterogeneous networks and efficient sharing of wireless networks such as WLAN.

For optimum use of multiple networks progress has been made in the IST-Daidalos project [4]. For efficient sharing of wireless networks a solution was reported in [34], that shared knowledge on QoS queue-lengths. A recent approach is that of IEEE 802.11e that uses a differentiated scheme with prioritized QoS classes including best-effort, video and audio. IEEE 802.11e also has non-mandatory extensions that can enforce the traffic constraints per terminal and QoS class.

5.2 Mobility and Sharing of WSANs

When a WSAN in a truck or on a body is used by applications over the Internet, it can temporarily lose network connectivity and may have to change to other networks as it moves. These mobility changes have impact on the bandwidth and latency of the information from WSAN, and on the reachability of the WSAN for remote configuration and actuation. Conflicts can arise when multiple applications try to send configuration and actuation commands to the WSAN. Another type of conflicts can arise when WSANs that use the same wireless resources move in each other's coverage area.

[1] Note that a mobile device can have multiple network interfaces that can be be connected to different networks simultaneously. Normally applications just use the default network interface, but they can use specific network interface for each connection they make.

Progress has been made in the IST-SENSEI project [18] with a.o. 6LoWPAN [30] and a binary web service protocol. In [14] a middleware solution is presented for sharing WSANs. In [13] approaches for mobility and sharing WSANs are further analysed.

5.3 Mobility and Sharing of Multimedia Sessions

A multimedia session is usually a combination of session control and multimedia streams between endpoints. For multimedia session mobility we therefore distinguish between changes in network interface attachment[1] of session control and multimedia stream endpoints. The latter enables splitting a multimedia session across multiple devices, e.g. move the video from your mobile to a nearby wall display and moving it back later. Multimedia streams can also be shared by multiple recipients when it is multicasted or otherwise duplicated, the challenge is to do this efficiently with realtime content to a dynamically changing and mobile group of users.

Progress has been made in the Freeband 4Gplus project [40] and IST-Daidalos project [3, 4]. A network-initiated method for splitting multimedia sessions is described in [2]. In [47] an approach is described for efficient personalized sharing of multimedia streams.

5.4 Service Platforms

A service platform offers a mobile terminal access to the network and services in heterogeneous networks. Federation between Service Platforms realises a service control layer that extends network and service usage to those of other service platforms. Challenges for service platforms are offering appropriate QoS and security while roaming, sharing your identity across networks and applications and enabling anonymous use of web and multimedia applications.

Regarding progress, mobility schemes for maintaining sessions were analysed in the Freeband 4Gplus project [40] and IST-Daidalos project [3, 4]. The IST-Daidalos project also worked on virtual identities across network and applications, with anonymity support.

5.5 Pervasive Service Platforms

Pervasive service platforms extend service platforms with composition of tailored and context-aware services, streams, context, into an pervasive application. The challenge for pervasive service platforms is to offer adaptability of the pervasive application to all sorts of changes such as environmental, the situation the user is in, the network attachment and the available bandwidth.

Regarding progress, the IST-Daidalos project [3, 4] proposes a pervasive service platform that offers this adaptability. In the context of the IST-Sensei project, a framework was created to enable comparing and combining pervasive communication architectures.

6 Conclusion

In this chapter we have described the main concepts involved in realtime mobile sharing, namely networks, sessions, services, mobility, federated service platforms, sharing and pervasiveness. We have noticed that the combination of these concepts can enable the IoT, IoS and IoM. We have also observed the dynamics of network attachment, multimedia sessions and context and how they can trigger and enable adaptations of pervasive applications. The bottom line is that sharing and mobility are intertwined, and the performance of realtime sharing and mobility handling impacts the efficiency and scalability of pervasive applications. We expect that for efficient realtime mobile sharing, support is required across the mobile device, the network and the service infrastructure.

Acknowledgements. Part of the work described in this chapter are results of the project 4th Generation Platform Launching Ubiquitous Services (4GPLUS), see http://4gplus.freeband.nl, and was supported in part by the Dutch Freeband Impulse Programme on Telecommunication Applications (http://www.freeband.nl) and is co-funded by the Dutch Ministry of Economic Affairs.

Part of the work presented in this chapter was partially funded by the EU project IST-2002-506997 DAIDALOS (Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services), and synthesizes the contributions of multiple project partners.

The work described in this chapter is based on results of IST FP6 Integrated Project DAIDALOS II (IST-2006-026943), which received research funding from the European Communities Sixth Framework Programme.

This chapter describes work in part undertaken in the context of the SENSEI project, Integrating the Physical with the Digital World of the Network of the Future (www.senseiproject.eu). SENSEI is a Large Scale Collaborative Project supported by the European 7th Framework Programme, contract number: 215923.

Apart from the research funding, the European Commission has no responsibility for the content of this chapter. The information in this chapter is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. 3GPP. IP Multimedia Subsystem (IMS); Stage 2. TS 23.228, 3rd Generation Partnership Project (3GPP)
2. Tuijn, J.A., Bijwaard, D.: Spanning a multimedia session across multiple devices. Bell Labs Technical Journal 12(4), 179–193 (2006)
3. Aguiar, R., Bijwaard, D., Jaehnert, J., Christ, P., Einsiedler, H.: Designing networks for the delivery of advanced flexible personal services: the daidalos approach. In: IST Mobile and Wireless Communication Summit (2004)

4. Aguiar, R.L., Sarma, A., Bijwaard, D., Marchetti, L., Pacyna, P.: Pervasiveness in a competitive multi-operator environment:the daidalos project. IEEE Communications Magazine 45(10), 22–26 (2007)
5. Aguiar, R., Pacyna, P.: Final Daidalos II global architecture. Deliverable DII-124, Daidalos consortium (October 2008)
6. Akyildiz, I.F., McNair, J., Ho, J.S.M., Uzunaliolu, H., Wang, W.: Mobility management in current and future communication networks. IEEE Network Magazine (August 1998)
7. Akyildiz, I.F., McNair, J., Ho, J.S.M., Uzunaliolu, H., Wang, W.: Mobility management in next generation wireless systems. Technical report, Georgia Institute of Technology (1999)
8. Open Mobile Alliance. Common definitions for OMA RESTful Network APIs, OMA-TS-REST_NetAPI_Common-V1_0, http://www.openmobilealliance.org/ (last visited July 2012)
9. Bargh, M.S., Bijwaard, D., Zandbelt, H., Meeuwissen, E., Peddemors, A.: Mobility management in beyond 3g environments. In: Proceedings of Wireless World Research Forum 9, WWRF 9 (July 2003)
10. Bargh, M.S., Zandbelt, H., Peddemors, A.: Managing mobility in 4g environments with federating service platforms (an overview). In: Proceedings of EVOLUTE Workshop, Evolute 2003 (2003)
11. Berger, A., Romascanu, D.: Power ethernet MIB. RFC 3621, IETF (December 2003)
12. Bhattacharyya, S., Ed, I.: An overview of Source-Specific multicast (SSM). RFC 3569, IETF (July 2003)
13. Bijwaard, D.J.A., Havinga, P.J.M., Eertink, E.H.: Analysis of mobility and sharing of wsns by ip applications. International Journal of Distributed Sensor Networks 2012, 923594 (2011)
14. Bijwaard, D.J.A., Kleunen, W.A.P., Havinga, P.J.M., Kleiboer, L., Bijl, M.J.J.: Industry: Using dynamic WSNs in smart logistics for fruits and pharmacy. In: Proceedings of SenSys 2011, Seattle, WA, USA, November 1-4, pp. 218–231. ACM, New York (2011)
15. Bosch, S., Marin-Perianu, R.S., Havinga, P.J.M., Marin-Perianu, M., Horst, A., Vasilescu, A.: Automatic recognition of object use based on wireless motion sensors. In: International Symposium on Wearable Computers 2010, Seoul, South Korea, USA, pp. 143–150. IEEE Computer Society (2010)
16. BSD. TinyOS, operating system designed for low-power wireless devices, http://www.tinyos.net
17. Callas, J., Donnerhacke, L., Finney, H., Shaw, D., Thayer, R.: Openpgp message format. RFC 4880, IETF (November 2007)
18. SENSEI consortium. SENSEI: Integrating the physical with the digital world of the network of the future, http://sensei-project.eu (last visited July 2012)
19. Deering, S.: Host extensions for IP multicasting. RFC 1112, IETF (August 1989)
20. Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: Network mobility (NEMO) basic support protocol. RFC 3963, IETF (January 2005)
21. Dierks, T.: The transport layer security (tls) protocol version 1.2. RFC 5246, IETF (August 2008)
22. Dutta, A., Jain, R., Wong, D., Burns, J., Young, K., Schulzrinne, H.: Multilayered mobility management for survivable network. In: Milcom, Vienna, Virginia (November 2001)
23. Eisenhauer, M., Rosengren, P., Antolin, P.: A development platform for integrating wireless devices and sensors into ambient intelligence systems. In: 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, SECON Workshops 2009, pp. 1–3 (2009)

24. XMPP Standards Foundation and Google. Xmpp technologies: Jingle, http://xmpp.org/about-xmpp/technology-overview/jingle/
25. Freier, A.O., Karlton, P., Kocher, P.C.: The SSL protocol version 3.0, http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt
26. Hofmeijer, T.J., Dulman, S.O., Jansen, P.G., Havinga, P.J.M.: AmbientRT - real time system software support for data centric sensor networks. In: Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, pp. 61–66. IEEE Computer Society Press (2004)
27. ITU-T. Packet-based multimedia communication systems, http://www.itu.int/rec/T-REC-H.323/en/
28. Johnson, D., Perkins, C., Arkko, J.: Mobility support in IPv6. RFC 3775, IETF (June 2004)
29. Kent, S., Seo, K.: Security architecture for the internet protocol. RFC 4301, IETF (December 2005)
30. Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. RFC 4919, IETF (August 2007)
31. Meratnia, N., van der Zwaag, B.J., van Dijk, H.W., Bijwaard, D., Havinga, P.J.M.: Sensor networks in the low lands. Sensors 10(9), 8504–8525 (2010)
32. Milonas, A.C.: Enterprise networking for the new millenium. Bell Labs Technical Journal 5(1), 73–94 (2000)
33. Misra, A., Das, S., Agrawal, P.: Application-centric analysis of IP-based mobility management techniques. Journal of Wireless Communications and Mobile Computing 1(3) (August 2001)
34. Peelen, B., Zivkovic, M., Bijwaard, D., Teunissen, H.: Supporting qos in broadband wireless and wired access. Bell Labs Technical Journal 8(2), 65–81 (2003)
35. Perkins, C.: IP mobility support for IPv4. RFC 5944, IETF (November 2010) (revised)
36. Perkins, C.: IP mobility support for IPv4. RFC 3344, IETF (August 2002)
37. Perkins, C., Johnson, D.B.: Mobility support in IPv6. In: Second Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom 1996, pp. 27–37 (1996)
38. Politis, C., Chew, K.A., Tafaz, R.: Multilayer mobility management for all-IP networks: Pure SIP vs. hybrid SIP/mobile IP. In: The 57th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Spring (2003)
39. Roach, A.: Session initiation protocol (SIP)-Specific event notification. RFC 3265, IETF (June 2002)
40. Romijn, W.A., Plas, D.-J., Bijwaard, D., Meeuwissen, E., van Ooijen, G.: Mobility management for SIP sessions in a heterogeneous network environment. Bell Labs Technical Journal 9(3), 237–253 (2004)
41. Rosenberg, J., Schulzrinne, H.: Session initiation protocol (SIP): locating SIP servers. RFC 3263, IETF (June 2002)
42. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: session initiation protocol. RFC 3261, IETF (June 2002)
43. Saint-Andre, P.: Extensible messaging and presence protocol (XMPP): core. RFC 3920, IETF (October 2004)
44. Schulzrinne, H., Rao, A., Lanphier, R.: Real time streaming protocol (RTSP). RFC 2326, IETF (April 1998)
45. Schulzrinne, H., Wedlund, E.: Application-Layer mobility using SIP. Mobile Computing and Communications Review (MC2R) 4(3), 47–57 (2000)

46. Stewart, R.: Stream control transmission protocol. RFC 4960, IETF (September 2007)
47. van der Gaast, S., Bijwaard, D.: Efficiency of personalized content distribution. Bell Labs Technical Journal 13(2), 135–145 (2008)
48. van Eijk, R., Brok, J., van Bemmel, J., Busropan, B.: Access network selection in a 4G environment and the roles of terminal and service platform. In: Proc. of Wireless World Research Forum 10, WWRF 10 (2003)
49. Wedlund, E., Schulzrinne, H.: Mobility support using SIP. In: 2nd ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM), Seattle, Washington (August 1999)
50. Jingle session transfer, http://xmpp.org/extensions/xep-0251.html (last visited July 2012)

Malware Detection and Prevention in RFID Systems

Harinda Fernando and Jemal Abawajy

Abstract. The threat that malware poses to RFID systems was identified only recently. Fortunately, all currently known RFID malware is based on SQLIA. Therefore, in this chapter we propose a dual pronged, tag based SQLIA detection and prevention method optimized for RFID systems. The first technique is a SQL query matching approach that uses simple string comparisons and provides strong security against a majority of the SQLIA types possible on RFID systems. To provide security against second order SQLIA, which is a major gap in the current literature, we also propose a tag data validation and sanitization technique. The preliminary evaluation of our query matching technique is very promising, showing 100% detection rates and 0% false positives for all attacks other than second order injection.

1 Introduction

RFID (Radio Frequency Identification) is a technology that has been around for a significant amount of time but only really received significant attention in the last decade or so. RFID is based on the concept of using electronic tags to identify and track physical objects, or people from a distance without physical or visual contact. The rapid evolution of RFID technology has led to the identification of a number of new types of RFID threats that were not an issue just a few years ago. One such major security concern for RFID systems is their recently identified vulnerability to tag based malware. In [1] the vulnerability of RFID systems to SQLIA (SQL Injection Attacks) were proven when the authors demonstrated how a fully functional RFID virus can be used to infect and spread via SQLIA. SQLIA refers to a specific type of malicious attack in which the data provided by the user is integrated into a SQL query so as to make that input be treated as part of the code rather than part of the input. This paper not only highlighted the possibility of SQLIA attacks compromising RFID systems, it also illustrated how the specific architecture of RFID systems makes RFID malware a real possibility. RFID malware spread by infecting new tags and databases. The nature of large networked RFID systems such as global supply chain management systems mean that if such a virus was deployed it would spread to hundreds of thousands of tags and hundreds of different systems very fast.

Harinda Fernando · Jemal Abawajy
Deakin University
e-mail: {hsf,jemal.abawajy}@deakin.edu.au

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 143–166.
DOI: 10.1007/978-3-642-34952-2_6

Fortunately all currently identified types of RFID malware are based on SQLIA [2]. Therefore if we can successfully defend against tag based SQLIA then we can defend against all currently know RFID malware.

While the vulnerabilities that leads to SQLIA are well understood they still persist because no truly effective techniques for detecting and preventing them still exist [3]. A number of different techniques have been proposed for SQLIA detection and prevention in web applications, but none of them have been completely effective. In addition the differences in the architecture of web applications and RFID systems mean that most of the approaches proposed for web systems do not work well with RFID systems. The increase in storage capacity of RFID tags and the drop in their prices have motivated more users to develop RFID systems that store more and more data on the tag itself for ease of access increasing the amount of potential systems that can be infected [4]. Hence removing the vulnerability of RFID systems to SQLIA and therefore malware is currently a very high priority.

Rather than developing a new solution from scratch we focused on three main priorities. (1) Understanding the differences in RFID systems and web based systems and therefore their vulnerabilities to SQLIA, (2) Identification of possible web based SQLIA defenses that can be successfully adapted for use in RFID systems and (3) Modification and improvement of those defense techniques so they are more suited for use in RFID systems. The technique proposed in this book chapter is a dual pronged defense mechanism for protecting RFID systems from tag-based SQLIA. The two techniques developed and presented are (1) Validation and sanitization of RFID based data to ensure that no "bad" data is used in generating dynamic queries and (2) Matching the structure of those dynamic queries with the legal structure for that query using simple string comparison.

In this chapter we also present the results of the evaluation of the security afforded by the proposed technique. Therefore the main contributions of the work presented here are:

1. Analysis and identification of differences in SQLIA in RFID systems and web systems and the identification of key requirements for any RFID defence techniques.
2. Review of current SQLIA detection and prevention techniques and identification of their weaknesses in relation to RFID networks.
3. Creation of a SQLIA defence mechanism for networked RFID systems that meets the identified requirements.
4. Evaluation of the proposed system to quantify its success rate.

The rest of this chapter is organized as follows. Section 2 explains SQLIA. Section 3 describes how malware can be mounted on RFID systems and analyses the main differences in RFID SQLIA and web based SQLIA. In Section 4 we present a literature survey of existing SQLIA detection approaches and analyze their weaknesses. Section 5 contains an overview of the proposed technique and we explain it in further detail in subsections. Section 6 explains how to develop the legal query structure strings that will be used in the query matching technique. We then present the results of the evaluation of our approach in section 7 and conclude in section 8.

2 SQL Injection Attacks

To allow full use of the data stored in databases most systems allow the users to input various parameters that are then used as part of an automatically generated SQL query. These queries are then forwarded to the database and the system carries out various processes based on the outcome of that query [5]. SQLIA are a unique form of malware that depends on injecting malicious SQL code (in the form of user input) into normal SQL queries and therefore the database. According to this definition a large number of different attacks possible databases such as tautologies, stored procedures, piggy backed queries, union attacks and attacks by errors all fall into the category of SQLIA.

Imagine a web page which takes in a user name and password and displays the users profile information. Once the user inputs a user name and password and clicks the submit button the page will run a script that dynamically generates a SQL query which contains both the user name and the password input by the user. The auto generated SQL command will be something like

```
SELECT * FROM users WHERE login = ‘usrnme’ AND password = ‘pswd’;
```

Where usrnme is the username input by the user and pswd is the password input by the user. Now imagine the attacker inputs “hsf’ or 1=1 –” as username and “gshg” into the web fields. Then the resulting query is:

```
SELECT * FROM users WHERE login = ‘hsf’ or 1=1 --’ AND password = ‘gshg’;
```

Once this query has been generated it will be sent to the database for execution. The database will interpret everything after the WHERE keyword as a conditional clause and everything after -- would be ignored as a comment. Because “or 1=1” part of the query is always true its inclusion into conditional clause causes the statement to always evaluate to true. Therefore on receiving the above query the database would return all details of all users to the attacker after executing this query. This is just one simple example of the wide range of possible SQLIA on databases [6].

3 Malware in RFID Systems

SQLIA can be used to attack RFID systems, additionally RFID malware attacks can be mounted using SQLIA as well. RFID malware is malicious code that when stored on RFID tags can propagate and when executed or used harm the overall system. Figure 1 shows a typical RFID system and illustrates how RFID malware can be used to infect the system via SQLIA using tags for the malicious data input. In typical RFID systems, the tags store data that is read by readers. This data is then forwarded by the readers to the middleware. The middleware uses the received data to build dynamic RFID queries (queries which have the tag data embedded into them). These queries are then forwarded onto the database. These dynamically generated queries can either retrieve data from the database or update

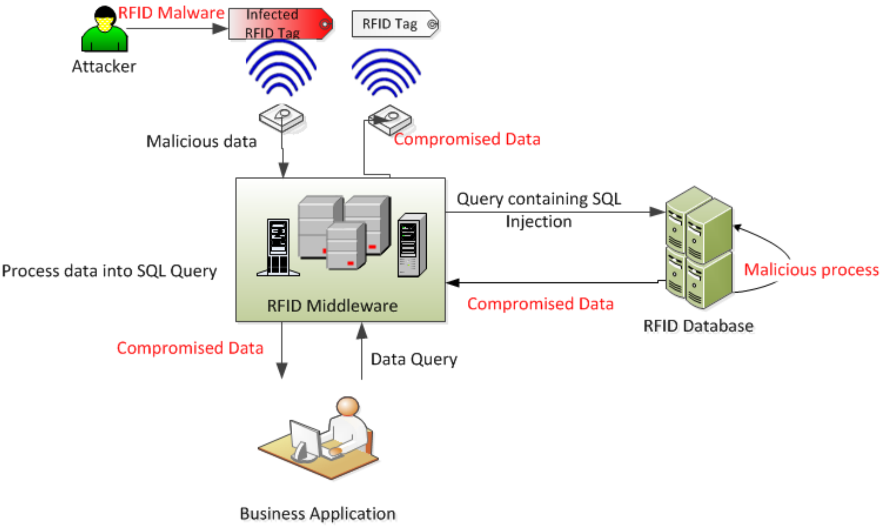


Fig. 1 Tag based malware in RFID systems

the existing data. Any results of the queries are sent back to the middleware. But unlike in web based systems where the data is sent back to the client that originally submitted the input data, in RFID systems the retrieved data is not forwarded to the tags. When queried by business applications the middleware retrieves the information as required from the database and forwards it to the business applications.

When an attacker wants to mount an SQLIA on this system, he saves the malicious data on the tag itself. When a reader polls a tag containing the malicious data, it will read and forward that data to the middleware. The middleware will then use that data to build dynamic SQL queries which are malicious and forward them to the database for execution [7]. These queries will command the database to carryout processes which compromise it or the data stored in it. In addition, properly written malicious data will act as a RFID virus and propagate to the database. Later on additional tags may be updated with the corrupted data stored in the database. If the malicious data is written correctly this will cause the recently updated tag to also become infected and it will in turn go on to infect and compromise other system's middleware and databases. This kind of RFID SQLIA malware can propagate extremely fast and infect a large number of tags and databases compromising them all [8].

3.1 Differences in Web Based and RFID SQLIA

Fortunately, attacking a RFID based system with a SQLIA is a lot more difficult proposition than attacking a web based system due to a number of reasons. In web based systems, because the dynamic SQL generation is carried out on the user machine, the data needs to be validated on the user's machine creating the need for distributed data validation. Additionally, in most big web sites there can be a very large number of constantly changing and expanding input sources in the form of interactive web pages. These web sites and pages can be built and maintained by external companies who are not too concerned about the security of the third party database they are accessing. This makes it extremely difficult to ensure the proper validation of all input into the system [6]. But in RFID systems the dynamic queries are generated at a single point in the RFID middleware. Therefore proper validation can be carried out by a single point in the middleware as well. In addition, unlike in web applications where the input can vary considerable the data received from RFID tags have a much more limited scope. Hence setting up data standards and verifying those standards are also significantly easier in RFID systems. Consequently, input validation and sanitization is much easier in RFID systems compared to web based systems. But RFID tag data cleaning still has its challenges as well. One main difference in web data and RFID data is that web data is normally input as discrete blocks with each data field being input separately. But in RFID tags the data is stored as one contiguous block and it is up to the middleware to actually identify each field and separate the data block into its component field. Therefore decisions on how the data will be stored on the tag and what formatting standards will be used have to be made and enforced if RFID malware is to be successfully defended against [9].

Another key feature of RFID systems is the limited amount of data stored on the tag and the limited access given to the tag. In web based systems, web clients may have full administrator access and may be able to input a vast number of different parameters for query generation, but in RFID systems tags are restricted to data input and even then the scope of the data is very limited. This allows the setting of very strict data standards, as the type, size and amount of data expected from tags are known in advance. This along with the single generation point for dynamic queries (the middleware) makes it much easier to validate and sanitize input data coming from the RFID tags relative to input data from web based systems. In addition, the numbers of different types of SQL queries that are automatically generated by the middleware are also much lower than the number of different queries generated in a web based system. Additionally, the limited number of queries and the fact that all those queries are set by the company itself and not outside companies makes the number of valid structures possible for the dynamically generated queries very low and easy to track.

Finally RFID tags are treated as simple data containers as opposed to web pages in web systems which are treated as input output devices. This means the tags can only provide raw data not queries. They also cannot perform or request for any other processes or data. RFID tags do not also receive data based on queries sent and cannot retrieve data from the backend databases. The tag updates are done by the middleware at its own choosing. Also, unlike in the web based systems where the queries are generated on an external client machine, the dynamic queries in RFID are all generated by internal servers. Therefore, it is not possible for potential attackers to gain access to the query structures beforehand making it that much more difficult to mount a SQLIA on a RFID system.

Table 1 Differences in web based SQLIA and RFID SQLIA

	Web-based Systems	RFID Systems
Query generation location	External (at clients computer)	Internal (in middleware)
Number of origin points for generated queries	Very large (large number of different web pages and web sites)	Single (only the middleware)
Number of different valid query structures based on input	Large and constantly changing	Small and fixed
Input output capabilities of attack origin	The web browser is both a input and output device letting the user input parameters and then view the results of the generated queries	The tags are treated as simple data containers. They hold data that can only be updated by the readers of the system. They cannot request for data and they do not receive any feedback
Data formatting and standards	Hard to set due to large number of different input points and input values possible	Can be easily set as tags contents are known well in adVance
Number of possible inputs for query generation	Very large and constantly increasing as more and more web pages and web sites are created which query the database	Small and known in advance
Access to query structures by attacker	Accessible as the query generation scripts must be sent to the attackers web browser	Not accessible by attacker as all query generation is done by middleware

Table 1 summarizes the major architecture differences of web based systems and RFID systems. These differences in architecture mean that some types of SQLIA's cannot be mounted on RFID systems. In total there are 9 different types of SQLIA that can be mounted [6]. Out of this 9 only 6 can be mounted on RFID systems. The reduced number of attacks is mainly because RFID tags do not receive results or error messages and therefore attacks based on receiving feedback from the system in response to the SQLIA are ineffectual on RFID systems. Overall it is much more difficult to mount SQLIA attacks on RFID systems and therefore protecting against them become much easier for RFID systems as well.

4 Related Work

Because SQLIA attacks depend on inputting invalid data to be used when building queries, ensuring the validity of the inputs will mitigate a majority of the possible SQLIAs. Defensive coding practices are simple SQLIA prevention techniques that revolve around ensuring that all accepted inputs are validated before being accepted [6]. Some of the best practices when it comes to defensive coding are input type checking and encoding of inputs.

The easiest data validation technique, input type checking consists of ensuring that the input data is type consistent with the expected data type for that value [10]. Because most SQLIA depends on inserting special characters or strings into inputs, these types of SQLIA can be blocked by this technique. In the same manner if a max and min length of specific inputs are known beforehand this information can be used in ensuring that additional characters have not been entered into the input [10]. As most SQLIAs require that the input is significantly larger than the expected length of the input this can catch a majority of the more complex SQLIA attempts. Another defensive coding technique: encoding of inputs consist of encoding the input in such a way as to ensure that the database does not mistake meta characters in the input for keywords, tokens or operators. Injection is often accomplished by tricking the system into accepting special characters embedded in string inputs as meta characters [6]. Hence, if the system can ensure that all string inputs are recognized as string and not meta characters attacks using these methods would fail.

Overall defensive coding techniques still remain one of the simplest and best ways with which to prevent SQLIA. Unfortunately, in web systems, defensive coding is prone to human error, mainly due to the fact that most developers do not remember to put in the required validation at all possible input locations, and therefore have been discarded as being too unsecure [6]. But in RFID systems, as all dynamic queries are generated by the middleware at a single point, it is sufficient to put input validation at the single query generation point in the SQL middleware. Therefore defensive coding practices are well suited for SQLIA prevention in RFID systems while remaining relatively simple and resource minimal.

Because defensive coding techniques proved to be unreliable in practical web application scenarios, researchers have proposed and developed a wide range of other techniques to detect and prevent SQLIA. The new query development paradigms proposed in SQL DOM [11] use encapsulation of database queries to provide a safe and reliable way to access the database. While these methods are secure they cannot be used for existing legacy systems without major redevelopment. They also require programmers to learn completely new development process based on the query development paradigm which is a lot of extra work. The intrusion detection system presented in [12] uses an automated machine learning technique trained using a set of typical application queries to try and detect SQLIA. This system first build models of valid queries and then uses pattern matching during run time to ensure that all received queries match a valid query model. The success of this approach is directly based on the quality of the training set used and a bad training set can result in a system with a large number of false positives and negatives. Therefore the security afforded by this system is always questionable.

SQL rand [13] is an instruction set randomization technique which uses a proxy based method and allows developers to create SQL queries using randomized instructions. This technique is based on cryptographic integrity check systems. Therefore, it not only places significant overhead on the system but its security is also fully dependent on the security of the secret key used in the randomization of the queries. Static code checking is a method by which the source code is checked for various weaknesses that make it vulnerable to SQLIA. The main drawback of this approach is that because only static code is analyzed it can only spot a limited number of weaknesses. For example the approach presented in [14] can only detect and prevent tautologies, while the approach presented in [15] can only spot weakness to incorrectly types inputs. This means that these types of approaches do not provide sufficient security for RFID systems.

Another detection technique, known as dynamic query pattern matching, consists of a hybrid of static code analysis and dynamic run time monitoring. In this technique the code is analyzed for weaknesses and all legal query patterns are identified and documented during the static phase. Then the identified query patterns are used to analyze and validate the SQL queries generated and submitted during the runtime monitoring phase. The major difference in the systems employing this technique is the method with which the legal query patterns are stored and the method with which the structure of the dynamic queries are compared to the legal query patterns identified during the static phase.

AMENESIA [16], SQLGuard [17] and SQL-Check [18] all use different variations of this basic technique. AMENSIA uses a web crawler to identify possible input sources (hotspots) for the system. This makes it impossible to be used in RFID systems as they don't have web inputs. Once all possible hotspots have been identified it uses the Java String analysis library to analysis the string operations carried out on each string of interest and deduct a non-deterministic finite automation that expresses all possible values the considered string can assume. Because the NDFA are an overestimate this technique may result in legal queries being mistaken for illegal queries. Both SQLGuard and SQL-Check take a different approach. They generate a parse tree to represent legal queries and compare them to the parse tree of the dynamically generated query. The difference is that SQL-guard the model is deduced automatically while the model for SQL-Check is developed by the programmer. Unfortunately both approaches use generated secret keys which must be kept secret and they both require the developer to use special intermediary libraries or to manually insert special markers in the code [3] which add unnecessary overhead as well as complications to the system. Additionally parse trees, especially for more advanced SQL queries, can be extremely complicated and therefore properly comparing two parse trees are generally a very complex task. Therefore, these approaches use considerably more resources than can be justified for use in RFID systems.

In general all current query pattern checking techniques have two weaknesses in common in the context of their use in RFID systems.

1. Un-needed complexity and computational overhead both in generating the legal query patterns and when comparing them with the patterns of dynamically generated queries.
2. Weakness in the query models due to the automated manner in which they are built resulting in false positives and negatives.

As the above review shows the techniques developed for web based SQLIA detection and prevention do not work very well in the simpler environment of RFID tag based SQLIA. A majority of the proposed approaches are unnecessarily complex and resource intensive while some others are simply not compatible with RFID systems due the differences in the architecture of the two systems.

While there has been a lot of work done in detecting and preventing web based SQLIA attacks very little work has been done on the same for RFID systems. The papers [19,20] discuss in detail how RFID systems can be subject to SQLIA attacks but present very little work on how to detect or prevent them. In [20] the authors mention the possibility of using input validation or attribute code technology to detect RFID based SQLIA but does not elaborate any further. In [19] the authors list some areas the database server administration must take into consideration when setting up the system but no further elaboration is done. In [8] the authors discuss the possibility of infecting databases with traditional viruses using RFID SQLIA. But as the basis for infection is still SQLIA, prevention of SQLIA will stop this type of attacks. Once again in this paper the authors list some rudimentary steps that can be taken to prevent this type of attack but no further elaboration is done on how they can be implemented or the exact mechanism behind these suggestions. In [21] the authors present a digital forensic system for tracking and identify SQLIA attacks on RFID. This approach is only useful after the fact and cannot be used to either detect possible SQLIA before they are executed or to actually prevent their execution and is therefore unsuitable as a security technique is better suited as a forensic technique.

Overall, key differences in RFID systems and web based systems mean that the solutions developed for web based systems do not directly translate to RFID systems too well: some uses automated systems that cannot be implemented within the confines of the RFID architecture while others employ unnecessarily complex systems that will place additional overhead on the system. In addition, very little work has been done in actually protecting RFID systems from SQLIA or SQLIA based malware. Because the architecture of RFID systems makes it possible to create and deploy RFID viruses based on SQLIA [1,8] it is imperative that a SQLIA detection and prevention method is developed for RFID systems taking into account the unique architecture features that differentiate them from web based systems.

5 Policy Based RFID Malware Detection and Prevention

In this section we propose a simple yet effective policy based two pronged system for the detection and prevention of RFID tag based SQLIA. The method we propose is based on existing SQLIA detection techniques which have been proposed for use in web based systems. But we have modified and optimized those methods significantly so that they are better suited for use in RFID systems. We have done these modifications and optimizations based on the following key features that differentiate web systems from RFID systems.

1. RFID tag data is highly structured and of lower volume compared web based inputs. That data is also integrated into dynamic queries at a single point in the system: the middleware. Therefore intercepting, validating and sanitizing that data is much easier compared to doing it in web based systems.
2. SQL queries are built by a single point in the middleware compared to web systems where they are generated on external client machines. Additionally the number of different types of dynamic queries is much less in RFID systems compared to web systems.

We describe the proposed approach as “policy based” because it requires that the developers set a number of policies concerning the valid tag inputs and legal query structures for the system. In the following section we will describe the proposed approach in detail.

5.1 Approach Overview

The proposed system (Figure 2) compromises of two different techniques: RFID tag data cleaning and query pattern matching. Each technique has two main phases: static analysis phase and runtime monitoring phase.

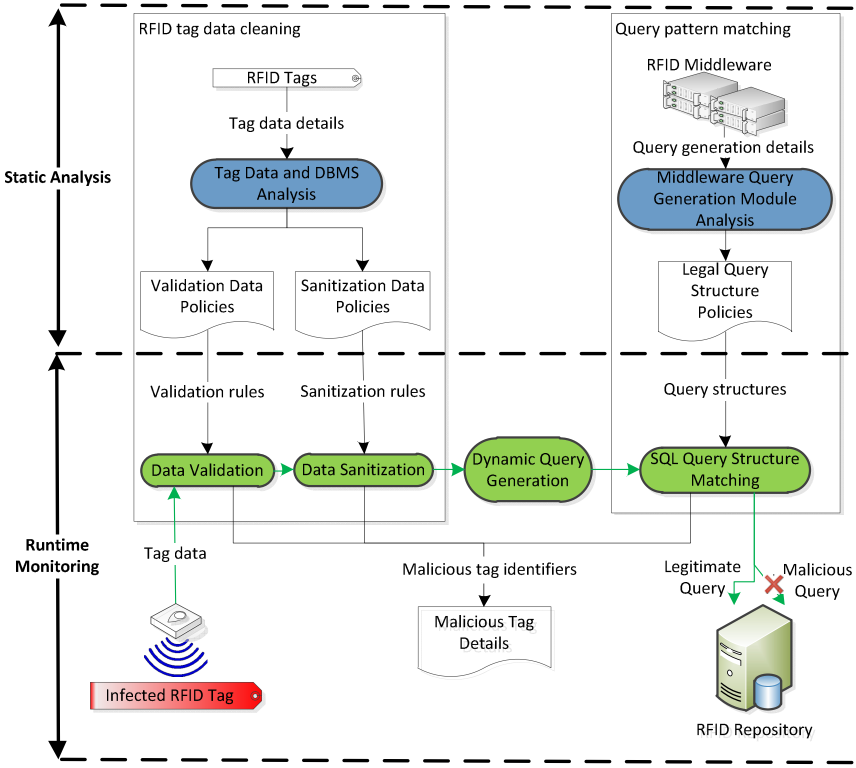


Fig. 2 Policy based RFID malware detection and prevention

The first technique creates data validation and sanitization policies during static analysis and enforces those policies during runtime monitoring. This ensures that only "clean" data is used in generating dynamic queries. The second technique is a SQL query pattern matching technique based on simple string comparison methods. This technique requires that the programmers define policies concerning the legal query structures (explained in section 6) during static analysis. The structure of the dynamically generated queries are then matched and validated, during runtime, against the legal query structures defined in the policies.

5.1.1 RFID Tag Data Cleaning

SQLIAs depend on inputting data in unexpected or unusual formations and structures to be successful. Therefore the root cause of SQLIA is insufficient input cleaning [6]. To ensure full RFID tag data cleaning we employ two different processes: Validation and Sanitization. Validation ensures that the data received from the external source adheres to pre-defined set of standards. Sanitization ensures that the data does not contain any "bad" data such as special characters or key words that have specific meaning to the system.

Data validation and sanitization has been dismissed as being unsuitable for securing web systems. But key differences in the architecture of web systems and RFID systems make it a suitable option for securing RFID systems. Additionally, because data validation and sanitization uses simple string comparison techniques their overhead is minimal ensuring high throughput and scalability. There are two distinct phases/steps to RFID tag data cleaning.

1. **Setting validation and sanitization policies** – This is carried out by a person with knowledge of both the contents of the tag and the DBMS used by the system. It is carried out during static analysis. This includes policies on tag data details such as length, type and formatting of the data. You also need to set policies on illegal keywords and characters for each data field.
2. **Tag data validation and sanitization** – This is done automatically by the system during runtime monitoring. It is done by identifying inputs that do not match the validation and sanitization policies.

5.1.2 SQL Query Pattern Matching

While data validation and sanitization is one of the simplest and most effective countermeasures to SQLIA there are methods by which it can be bypassed [6]. By using alternate encoding mechanisms as well more complex SQLIA, attackers can bypass the data validation and sanitization process. To ensure security against these types of attacks we propose a second security mechanism. This approach takes into account the structure of legal SQL queries for the system and compares it to the structure of the queries dynamically generated using RFID data. Our query pattern matching mechanism takes advantage of the fact that SQL injection changes the structure of the query to identify potential SQLIA and prevent them from being sent to the database.

The proposed approach is a simple and computationally minimal query pattern matching technique which employs string comparisons and is sufficient for protecting RFID systems. It is also easy to develop as most systems already include simple string comparison functionality. Evaluation shows that it provides stronger or equivalent protection to what is offered by other query pattern matching systems such as [13,16,17] when implemented in the specific architecture present in RFID systems. The simplicity of the technique is possible because the query generation is done on the middleware which has access to the parse function calls of the database and not on external web pages run on client machines. The query pattern matching approach technique consists of two steps:

1. **Defining legal query structure policies** – done during the static analysis phase, this consists of giving each different query a unique identifier and defining their query structure in a format available and understandable by the middleware. This process is explained in detail in section 6. In our technique this is manually done by the developer who codes the dynamic query generation code. If any additional queries are added these will also need unique identifiers and query structure policies.
2. **Query structure matching** – this consists of extracting the structure of dynamically generated queries by parsing (but not executing) it and seeing if it matches the legal query structure for that type of query as defined in the legal query structure policies.

5.2 Static Analysis

During static analysis the first task is the creation of the validation policies which contain rules about the structure of tag data. For this, first data must be stored as separate values rather than one long contiguous block on the RFID tags. This can be done by first identifying all data fields that will be stored on the tag and by ensuring that each field has a specific use. Then a method with which to identify each field needs to be developed. This can be done by giving each field a unique identifier, whether it is a number or name. E.G:- ID, Product Name, etc. Next key data features that can be used for validation of must be identified. Normally these are features such as data-type, max length, min length etc. Finally the values for each of the data features must be identified for each field and stored in a form which is available to the middleware. This is information such as data-type, max length and min length of each specific data field. Table 2 shows an example of a validation policy table for a simple system.

Next the data sanitization policies must be set (Figure 3). As we have already identified and named/numbered all possible fields that will be stored on the tag, now we must create the sanitization rules for each of those fields. There are two main requirements to fully sanitize data. Data must be clean of illegal specials characters (=, *, ; " etc) and data must be clean of any illegal keywords, tokens or function names. (Keywords and tokens are defined here as strings or parts of string that have specific meaning to either the DBMS or other software that use data from the database). To do this for each identified field first analyze if any special characters are not allowed to be contained in that field. If so decide which characters are not allowed and store them in form available to the middleware.

Next for each identified field identify if any specific words are not allowed in that field. If yes decide which "bad" data (key words, special characters and reserve words etc) are not allowed and store them as the sanitization data which is available to the middleware. Table 3 shows an example sanitization policy table.

Table 2 Example validation policy table

Field ID	Field Name	Data Type	Max length	Min length	Structure of the data field	Min possible value of data	Max possible value of data
1	Name	Alphabet-ic	30	5	String	N/A	N/A
2	Manu-facture date	Date	10	10	--/--/----	01/01/2 000	Current date
3	Batch number	Numeric	10	10	__-__-_	000-000-0000	999-999-9999
4	Price	Numeric	8	4	Number	0	1000
5	Delivery Address	Alphanu-meric	30	30	String	N/A	N/A

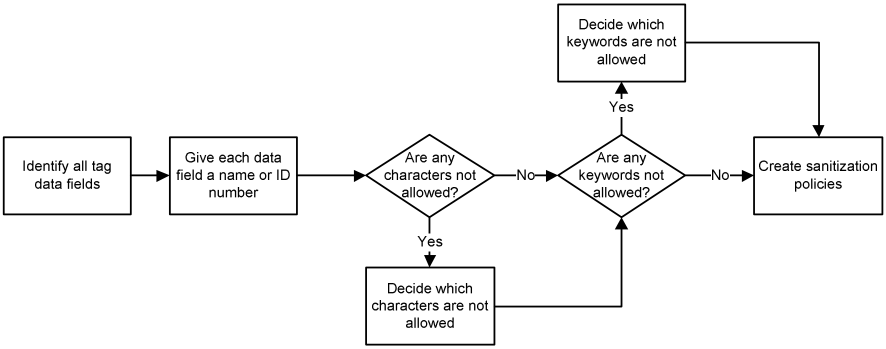

Identify all tag data fields
Give each data field a name or ID number
Are any characters not allowed?
No
Yes
Decide which characters are not allowed
Are any keywords not allowed?
Yes
Decide which keywords are not allowed
No
Create sanitization policies

Fig. 3 Sanitization policy creation

Finally, during the static analysis phase, legal query structure policies must be created for each query that will be generated by the middleware. For this, first, all possible query types that incorporate RFID tag data and are dynamically generated by the middleware need to be identified and given a unique identifier. Then, the final syntax for each identified query must be defined and the legal query structure must be created (further explained in section 6). RFID systems have relatively little dynamically generated queries containing tag input compared to web systems. Additionally all the queries are developed internally by the company who develops and runs the middleware and database. Therefore, we recommend that the programmer who develops the query generation software also define the legal query structure for each query manually. This has the twin advantages of minimizing the coding required and ensuring the correctness of the developed query models without fear of over compensation inherent in models developed by automated systems. Once the legal query models have been developed, they must be saved along with the corresponding unique identifier of the SQL query for that model.

Table 3 Example sanitization policy table

Field ID	Name	Characters not allowed	Character instances not allowed	Keywords not allowed	Keyword instances not allowed
1	Name	YES	/ : * = - . ( ) ! > < ;	YES	IF, OR, SHUT, NULL
2	Manufacture date	YES	: * = . ( ) ! < > ;	N/A	N/A
3	Batch number	YES	/ : * = . ( ) ! < > ;	N/A	N/A
4	Price	YES	/ : * = - ( ) ! < > ;	N/A	N/A
5	Delivery Address	YES	: * = - ( ) ! < > ;	YES	IF, OR, SHUT, NULL

To ensure continuing strength of the policies we recommend that the static analyses phase be an ongoing process with the rules being constantly updated as new functionality, dynamic queries and programs that access the database are added to the overall system.

5.3 Runtime Monitoring

Once all required policies have been identified during the static phase the system enters the run time monitoring phase. During this phase the system reads data from the tags. When data is retrieved from the RFID tags it arrives at the middleware as a single stream of tag data (TD). Before the middleware can apply the validation and sanitization policies it must first identify and separate each individual field (td_i where i = 1 to n) in the stream. The field identifier (i) is then used to extract the validation policies (td_iFF) for that field from storage. Then for each individual field of data (td_i) the data feature values such as max length, min length and data-type (td_iv_j) must be extracted by analyzing the separated data fields. Then those extracted feature values must be matched against the values stored in the validation data (td_iff_j) to see if adheres to the proper data standard. If the values match the policies then the data are passed on for sanitization else it's rejected and the tag is identified as being malicious in the malicious tag details.

Then the system needs to sanitize that data to ensure that it does not contain known “bad” data as defined by the sanitization policies. The sanitization function checks the data for illegal keywords/characters exist in the inputs as defined by the sanitization policies. To do this it takes the validated data (td_i where $i = 1$ to n) and retrieves the corresponding illegal keyword/character data (td_iK) for that field from the sanitization data. It then analyses td_i to see if any illegal tokens/keywords are included in that data. If any illegal tokens/keywords exist that data is rejected and the tag is marked as malicious. If it passes sanitization td_i is handed over to the query generation system. Algorithm 1 presents the algorithm for data cleaning based on preset policies.

TD - Tag data as a BLOCK
td_i - Tag data (TD) divided into n different fields with each field identified by i
td_iFF - The feature values (max length, data type etc) set for the tag data field td_i
td_iff_j - The allowed value of feature j for tag data field i
td_iv_j - The actual value of feature j for tag data field i
td_iK - The illegal keywords/characters for tag field i

Algorithm 1. RFID tag data cleaning algorithm

```
INPUT: TD, $td_iFF$, $td_iC$, $td_iK$
OUTPUT: Validated and sanitized RFID tag data
BEGIN RFID tag data cleaning
    Receive tag data (TD) from a reader
    Split TD into the separate fields ($td_i,\ldots,td_n$)
    FOR EACH ($td_i$ where i = 1 to n) DO
        Identify the data field using i
        Retrieve the feature values $td_iFF = [td_iff_{1,\ldots,} td_iff_m]$ for $td_i$
        FOR EACH ($td_iff_j$ where j = 1 to m) DO
            Extract the corresponding values $td_iv_j$ from $td_i$
            IF ($td_iv_j$ does not match $td_iff_j$) THEN
                Reject data
                Mark that tag as suspicious
            ENDIF
        ENDFOR
        Retrieve the illegal keyword data $td_iK$ for $td_i$
            IF (any keywords in $td_iK$ exist in $td_i$) THEN
                Reject data
                Mark that tag as suspicious
            ENDIF
        Forward $td_i$ to SQL query engine
    ENDFOR
END RFID tag data cleaning
```

The next step during runtime is comparing the structure of the dynamically generated queries with the structures in the legal query structure policies. To do this the query matching modules must first receive a generated query (GQ) and the associated identifier (ID) from the query generation module. When the query is received the module calls the parse function of the DBMS and inputs GQ as a parameter. The DBMS parses that query (but does not execute it) and returns the resulting parsed query (GQ_p) back to the query matching module. The module then uses GQ_p to generate the actual query structure (QS_a) of GQ. Then the module uses ID to retrieve the legal query structure policies (QS_l) corresponding to GQ. Finally it compares QS_l with QS_a. If the two does not match the query is identified as a SQLIA and rejected. Otherwise it's forwarded to the database for execution. The algorithm for this process is presented in Algorithm 2.

GQ – Dynamically generated query
ID - Unique identifier that associates GQ with the legal query pattern
GQ_p – GQ after is has been parsed by the database
QS_a – Actual query structure of GQ as extracted from GQp
QS_l – Legal query structure for GQ as defined by developer

Algorithm 2. Query structure matching algorithm

```
INPUT: GQ, ID, $QS_l$
OUTPUT: validated QS
**BEGIN** Query structure matching
    Receive GQ and corresponding ID from middleware
    Submit GQ to a parse function of the DBMS
    Receive $GQ_p$ as output of parse function
    Generate $QS_a$ by removing literal tokens from the parsed query $GQ_p$
    Use ID to retrieve $QS_l$ from storage
    IF ($QS_l$ != $QS_a$) THEN
        Reject query
    ELSE
        Submit query to DBMS for execution
    ENDIF
END Query structure matching
```

6 Query Structure Creation and Comparison

When developing a query structure format to be used in identifying legal queries we used the concept of tokens to decompose the query into its different constituent parts. We then use those different parts to develop a modular string based query structure for any given SQL query. This query structure indicates the logical structure of the query but removes any user inputs. Our technique is based on two important features present in all dynamic RFID queries

- Queries of the same type generated using different tag data will differ only in the user input values in the query.
- The input from the tag will not change the overall logic and structure of the resulting dynamic queries. In other words the user input from the tag is not meant to have any SQL statements or sub statements.

The main difference in our approach to other existing query structure matching techniques is query structure format and the method of comparison used. Unlike a lot of other similar systems which converts the tag structure into a XML document for analysis [22], or employ complex parse trees and compare them [17,18], or build finite automata for comparison purposes [16,23] we build string structures and use a simple string comparison to carry out the comparison.

6.1 Query Tokenization

The first step in creating legal query structures is separating the queries into the different types of tokens that it's composed of. In our approach tokens are defined as individual string parts and can be one of five main types: Keywords, Symbols/Operators, Identifiers, Literals and Comments

- **Keywords:** These are words that have specific meaning to the DMBS (SELECT, FROM, INSERT, WHERE and predefined functions such as AVG(), SUM(), CONCAT())
- **Symbols/Operators:** These are either single or compound symbols that have a specific meaning to the DBMS (+, =, ',;, etc)
- **Identifiers:** These are words that identify specific database components (table names, column names and user defined variables)
- **Literal:** These are bits of code that indicate the literal value of an item (e,g:-scott, 23.56, 12/07/1982). In our system variables are also considered literals as the variable itself will be replaced by a literal value when the query is dynamically generated by the middleware.
- **Comments:** Extra code that do not have any meaning to the database and is therefore ignored.

The first three types of tokens are important for the logic and structure of the query. Literals are only user input, and these have no effect on either the logic or the structure of the query. The last type of token; comments, is used by the programmers to makes notes for future use and has no effect on the actual query. Legitimate RFID tags are not meant to contain any of the first three types of tokens or comments and only contain actual values, or in other words the literals, contained in the query.

Keeping this in mind we begin developing SQL query structures as follow: The first step in creating the SQL query structures is to break the query down into its component string compromising of words and symbols and identify the type of each token each substring is. This is done as explained in the next section.

6.2 Query Structure Policy Generation

Imagine a RFID middleware system which takes as input the tag ID and the product it's attached to as input from the tag and saves that data to a database table. The resulting query for this process is as follows:

INSERT INTO product (tag_id, product_name) VALUES ('tagid', 'productname');

In the above query 'tagid' and 'productname' are string variables which are read from the RFID tag. The tokenized version of the above query, with each bit string separated and identified, would be as shown in figure 4:

Dark blue – keywords
Orange – identifiers
Green – operators/symbol
Red – literals

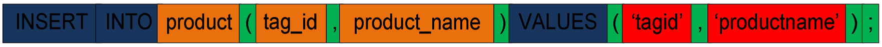


Fig. 4 Tokenized query

Now if we strip the literals (the red background) and replace them with "?" as a marker and remove comments from the query we get the common query structure for all dynamic queries generated for inputting product details into the product table based on tag input which is as follows (figure 5):

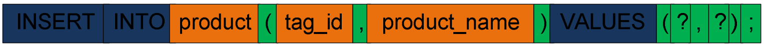


Fig. 5 Tokenized query with literals removed

By replacing the literals with "?" we ensure that the structure does not take into account the changing values for each different query, allowing the tag input to change as required. By keeping the first three types of tokens we ensure that the structure contains all the data concerning the query logic and structure, allowing for the logic of the dynamic queries to be validated. We discard the comments as they play no part in the query and are not used by the database.

Once the query structure is identified by stripping all the literals and comments, it is then converted into all lowercase (as SQL is not case insensitive) and saved as a string along with the unique identifier for that particular query type. In the same manner the query structure for all dynamically generated queries must be identified and analyzed. Table 4 shows a table containing some example query structures.

Table 4 Legal query structure table

Query Identifier	**Query Structure STRING policy**
1	insert into product (tag_id, product_name) values (?,?);
2	select * from product where tag_id = ?;
3	update product set stock_count = ? where product_id = ?;

6.3 Runtime Query Matching

During runtime the query structure matching module receives the generated query and the query identifier which indicates which query pattern policy it should match. At this point the module first parses the received query and uses the resulting parse information to identify the different token types and strip it of any literal tokens and replace them with "?". It then uses the received ID to retrieve the legal query structure string for the received query and compares it with the query structure string it generated. If the two match the query is validated and sent to the database for execution if not its identified as containing a SQLIA and discarded.

Our technique uses a much simpler method of comparison compared to the methods used by other techniques such as parse trees, XML documents or finite. Most systems and programming languages have built in string comparison and manipulation controls which make implementing this kind of comparison easier than the more complex custom types. Additionally string comparison is also a much quicker and less resource intensive comparison method compared to the other more complex methods used.

7 Security Evaluations

The goal of the evaluations carried out and presented in this section is to test the effectiveness of the malware detection approach presented in this chapter. The security afforded by data cleaning is directly tied to the completeness and strength of the data cleaning policies. Therefore it is nearly impossible to quantify the security this approach can provide without knowing exactly the system details and the policies set by the developers for data validation and sanitization. Hence in this section we will only discuss the additional security afforded by the RFID tag data cleaning approach at an analytical level. Additionally we will also present the results of the testing carried out for the query structure matching technique.

7.1 RFID Tag Data Cleaning

Input validation is one of the simplest and most effective ways of preventing the simpler types SQLIA [6]. Unfortunately because the security it affords depends on the strength of the data cleaning rules provided for the system. Therefore most data checking/cleaning techniques fail, not due to a flaw in its concept but due to weaknesses or incompleteness in the rules developed for them [6]. We have already analyzed the architecture of RFID systems and determined that RFID systems are better suited for input validation techniques as a security measure against tag based SQLIA than web based applications.

To minimize possibility of weak or incomplete data cleaning policies we have used two different data cleaning techniques based on two different core concepts: validation which is based on the concept of white listing and sanitization which is based on the concept of black listing. The use of this combination enhances the security afforded by the proposed technique by ensuring that more variables and factors are taken into account by the people who develop the data cleaning rules. Additionally the validation and sanitization policy creation process we have developed and presented in section 4.2 has been specially designed to ensure the strength and completeness of the policies created. This is done by ensuring that the policies take each separate data field into account. The policy creation process also ensures that multiple features and attributes of each data field are used in creating the validation and sanitization policies leading to much stronger and through policies.

In addition to its simplicity and its effectiveness against simpler SQLIA the other reason we use validation and sanitization in our technique is protection against second order SQLIA. Unlike other SQLIA, second order injection attacks do not change the structure of the dynamically generated query [2]. Therefore SQLIA detection techniques that rely on query structure/pattern matching, such as [13,15,16,11] and even the approach we have proposed, are ineffective against these types of attacks. Therefore second order injection still remains a very prominent threat to the security of most web and RFID applications. But, because RFID tag data cleaning does not depend on the query structure, but the rather the format and content of the input data, it can still be used to spot possible instances of second order injection in RFID tags. Therefore in our proposal we have included the data validation and sanitization technique in addition to the query matching technique to ensure that there is some protection against second order SQLIA. But, the protection provided by data validation and sanitization techniques is only as strong as the data validation and sanitization rules/policies set for it by the developers. Therefore it is imperative that the analyses carried out in the static phase are complete and the rules created are comprehensive. It's also important that the developers take into account the different other systems that will be accessing the RFID database and build the tag data cleaning rules with their weaknesses in mind.

7.2 Query Structure Matching

To carry out a thorough evaluation all three main types of dynamically generated SQL queries possible in RFID systems (SELECT, UPDATE, INSERT) had to be tested. Therefore we developed a number of queries of each type ranging from simple to complex and developed the legal query structure for each query.

To evaluate our technique we used two programs. One is the freely available demo version of the General SQL parser (GSP Demo) downloadable at http://www.sqlparser.com/download.php. The other was a simple string comparison program written by us. For parsing of the dynamic queries we used the pretty print facility of the GSP Demo. All tokens except comments, strings and numbers were set to show with green font color. Comments were blue while strings and numbers were red. Once the query was parsed we replaced all red text (literals) with "?" and deleted all blue text (comments). The resulting string was then compared with the legal query structure using a simple string comparison program we wrote. The program takes the dynamic query and strips any newline characters and any multiple spaces replacing them with single spaces. It then runs a single string comparison to compare result with the legal query structure input at the bottom of the program.

Table 5 Evaluation results

	Select Queries tested(Detected)	Update Queries tested(Detected)	Insert Queries tested(Detected)	Total (Detected)
Tautologies	21(21)	21(21)	N/A	42(42)
Union query	18(18)	6(6)	12(12)	36(36)
Piggy backed queries	15(15)	15(15)	15(15)	45(45)
Alternate encodings	12(12)	12(12)	12(12)	36(36)
Commenting queries	2(2)	5(5)	1(1)	8(8)
Total	68(68)	59(59)	40(40)	167(167)

Not all types of SQLIA can be mounted on RFID systems. Therefore when testing our system we only tested the types of attacks possible on RFID systems and ignored SQL attacks such as timing attacks, inference and illegal/illogical queries. Table 5 shows the breakdown of our testing process and the results obtained when testing malicious queries.

For all types of queries and all types of SQLIA types tested our query structure matching technique was able to identify SQLIA with 100% efficacy. In addition during the testing process we also tested around 120-130 legal queries. All legal queries were allowed by the technique with a 0% false positive rate. Even though the testing was limited to around 300 queries in total and carried out at logical level rather than implementation level the 100% detection rate and 0% rate in false positives are very promising.

In table 6 we compare the detection results of our approach against the security by some other SQLIA detection techniques. Please note the results do not take into account second order injection attacks. As the results show our approach is on par if not better than the best of the other approaches that are available in literature. Unfortunately, for three of the proposed techniques the original authors did not present their evaluation results. These papers are indicated with “Results not presented by original authors” in the table.

Table 6 Security comparison table

	Detection rate	False Positive rate
Proposed Approach	100%	0%
AMNESIA [16]	100%	0%
SQLCheck [18]	100%	0%
SQLGuard [17]	Results not presented by original authors	Results not presented by original authors
SQLrand [13]	Results not presented by original authors	Results not presented by original authors
Tautology-checker [14]	< 100%	Not Available
CANDID [24]	100%	0%
SQLDOM [25]	Results not presented by original authors	Results not presented by original authors

Table 7 Comparison of SQLIA detection techniques

	Tautology	**Union**	**Piggy Backed**	**Alternate encoding**	**Commenting**	**Second order injection**	**Notes**
Proposed Approach	1	1	1	1	1	1	Uses simple string comparison
AMNESIA	1	1	1	1	1	0	Uses NDFA which may over or under estimate the structures
SQLCheck	1	1	1	1	1	0	Uses parse trees for comparison and secret keys which increase system overhead
SQLGuard	1	1	1	1	1	0	Uses parse trees for comparison and secret keys which increase system overhead
SQLrand	1	1	1	1	1	0	Use high over cryptographic techniques
Tautology-checker	1	0	0	0	0	0	Only works for tautologies
CANDID	1	1	1	1	1	0	Uses a dynamic method to guess the programmer intended query structure
SQLDom	1	1	1	1	1	0	Needs a custom set of classes be built for each database schema High overhead

Additionally, as table 7 shows our technique is the only technique that actually has a possibility of detecting and preventing second order injections (given strong enough data cleaning rules). It is also one of the simplest approaches available as its uses only simple string comparison for both techniques compared to the more complex techniques used by other systems.

8 Conclusions and Future Work

In this chapter, we presented a simple but secure approach for detecting and preventing tag based RFID SQLIA. The overall technique consists of two different methods. The first method is a simple validation and sanitization technique for RFID tag data which is based on data validation and sanitization policies created by the developers based on tag data. This technique prevents ‘bad’ data from being used while building dynamic queries and is effective against second order injection attacks. The second method is a SQL query structure matching technique which uses simple string comparisons to identify possible SQLIA. This technique has the advantage of protecting against all other SQLIA types possible on RFID systems while being simpler than other proposals that use more complex matching techniques such as parse tree validation [17] or query randomization [13] .

The testing of the query structure matching method yielded very positive results. We tested all possible types of dynamic queries that may be generated in RFID systems with all possible types of attacks that can be mounted on those systems. In all more than 300 queries were tested with around 170 attacks and around 130 legal queries. The testing showed a detection rate of 100% and false positive rate of 0%. Our approach (specifically the validation and sanitization technique) was specifically designed to protect against second order injection attacks on RFID systems. This type of SQLIA cannot be detected by any query matching system. This, to the best of our knowledge, is the only work in current literature that has looked at detecting and preventing this type of SQLIA in RFID systems. The main weakness of the proposed techniques is that the security against second order injection relies heavily on the data cleaning policies. To ensure the strength and completeness of the data cleaning policies we have a developed and presented a data cleaning policy creation methodology that is highly structured. This methodology ensures that policies created by following it takes a large number of different attributes into account and is as complete as possible.

While the logical testing of our query matching technique was promising more thorough testing is required in actual deployed environment. Therefore as future work we plan to continue this work by implementing and testing our approach as a prototype and testing it in a runtime environment to evaluate its performance. We also plan on carrying out testing of our validation and sanitization technique by looking at real life RFID systems and developing realistic policies for the validation and sanitization of the tag data of those systems and then testing the security afforded by those policies against SQLIA.

References

1. Rieback, M., Simpson, P., Crispo, B., Tanenbaum, A.: RFID malware: Design principles and examples. Pervasive and Mobile Computing 2(4), 405–426 (2006)
2. Fernando, H., Abawajy, J.: Securing RFID Systems from SQLIA. In: Xiang, Y., Cuzzocrea, A., Hobbs, M., Zhou, W. (eds.) ICA3PP 2011, Part II. LNCS, vol. 7017, pp. 245–254. Springer, Heidelberg (2011)
3. Amirtahmasebi, K., Jalalinia, S.R., Khadem, S.: A survey of SQL injection defense mechanisms. In: 6th International Conference for Internet Technology and Secured Transactions, London, UK, November 9-12, pp. 1–8. IEEE (2009)
4. Schuster, E.W., Allen, S.J., Brock, D.L.: Global RFID. Springer, Berlin (2007)
5. Kindy, D.A., Pathan, A.K.: A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques. In: IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore, June 14-17, pp. 468–471 (2011)
6. Halfond, W., Viegas, J., Orso, A.: A classification of SQL-injection attacks and countermeasures. In: International Symposium on Secure Software Engineering. Citeseer (2006)

7. Rieback, M., Tanenbaum, A., Crispo, B.: RFID Malware: Truth vs. Myth. IEEE Security and Privacy 4(4), 70–72 (2006)
8. Suliman, A., Shankarapani, M., Mukkamala, S., Sung, A.: RFID malware fragmentation attacks. In: International Symposium on Collaborative Technologies and Systems, Irvine, CA, pp. 533–539. IEEE (2008)
9. Fernando, H., Abawajy, J.: A RFID Architecture Framework for Global Supply Chain Applications. In: 11th International Conference on Information Integration and Web-based Application and Services, Kular Lampur, Malaysia. ACM (2009)
10. Brabrand, C., Møller, A., Ricky, M., Schwartzbach, M.I.: Powerforms: Declarative client-side form field validation. World Wide Web 3(4), 205–214 (2000)
11. McClure, R.A., Krüger, I.H.: SQL DOM: compile time checking of dynamic SQL statements. In: 27th International Conference on Software Engineering, Missouri, USA, pp. 88–96. ACM (2005)
12. Valeur, F., Mutz, D., Vigna, G.: A Learning-Based Approach to the Detection of SQL Attacks. In: Julisch, K., Kruegel, C. (eds.) DIMVA 2005. LNCS, vol. 3548, pp. 123–140. Springer, Heidelberg (2005)
13. Boyd, S.W., Keromytis, A.D.: SQLrand: Preventing SQL Injection Attacks. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 292–302. Springer, Heidelberg (2004)
14. Wassermann, G., Su, Z.: An analysis framework for security in Web applications. In: First FSE Workshop on Specification and Verification of Component-Based Systems, p. 70 (2004)
15. Gould, C., Su, Z., Devanbu, P.: JDBC checker: A static analysis tool for SQL/JDBC applications. In: 26th International Conference on Software Engineering, pp. 697–698. IEEE (2004)
16. Halfond, W.G.J., Orso, A.: AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks. In: 3rd International ICSE Workshop on Dynamic Analysis, MO, USA, pp. 174–183. ACM (2005)
17. Buehrer, G., Weide, B.W., Sivilotti, P.A.G.: Using parse tree validation to prevent SQL injection attacks. In: International Conference on Software Engineering and Middleware, pp. 106–113. ACM (2005)
18. Su, Z., Wassermann, G.: The essence of command injection attacks in web applications. In: 33rd Annual Symposium on Principles of Programming Languages, pp. 372–382. ACM (January 2006)
19. Sulaiman, A., Mukkamala, S., Sung, A.: SQL infections through RFID. Journal in Computer Virology 4(4), 347–356 (2008)
20. Zhang, Q., Wang, X.: SQL Injections through Back-End of RFID System. In: International Symposium on Computer Network and Multimedia Technology, pp. 1–4. IEEE (2009)
21. Kyaw, A.K.: Digital Forensics in small devices: RFID tag investigation. AUT University, Auckland (2011)
22. Das, D., Sharma, U., Bhattacharyya, D.: An Approach to Detection of SQL Injection Vulnerabilities Based on Dynamic Query Matching. International Journal of Computer Applications IJCA 1(25), 39–45 (2010)
23. Gould, C., Su, Z., Devanbu, P.: Static checking of dynamically generated queries in database applications. In: 26th International Conference on Software Engineering (2004)
24. Bisht, P., Madhusudan, P., Venkatakrishnan, V.N.: CANDID: Dynamic candidate evaluations for automatic prevention of SQL injection attacks. ACM Transactions on Information Systems Security 13(2), 1–39 (2010), doi:10.1145/1698750.1698754
25. McClure, R.A., Kruger, I.H.: SQL DOM: compile time checking of dynamic SQL statements. In: 27th International Conference on Software Engineering, May 15-21, pp. 88–96 (2005)

Conceptual Model of Business Services Availability vs. Interoperability on Collaborative IoT-enabled eBusiness Platforms

Natalia Kryvinska and Christine Strauss

Abstract. Modern business background can be seen as a logical effect of eBusiness, advanced integrated networks, Internet of Things (IoT), and software services. In such an environment, the IoT-centered application deployment and delivery models have revolutionized the way businesses interact, collaborate and transact with customers, suppliers, partners, employees and shareholders. Hence, with wide deployment of the distributed inter-enterprise Service Delivery Platforms (SDPs) over the Internet, there is an urgent need to understand and solve service traffic issues of the fast evolving architectures. Accordingly, the purpose of our work is to develop a conceptual model for performance analysis of software services availability vs. interoperability in order to facilitate enterprises to attach their customers more tightly by an effective service delivery, which in turn optimizes business processes at different steps. Thus, we introduce and deliberate in this chapter a hypothetical model for the performance analysis of services availability and interoperability on the IoT-enabled inter-enterprise SDPs. We also figure and analyze imperative performance features of the model. The related open issues and future work are briefed correspondingly.

1 Introduction

In order to expose the role and the meaning of the IoT (Internet of Things) in/for the eBusiness and business services delivery, we have used the following definitions of it:

– "A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities.

Natalia Kryvinska
Secure Business Austria (SBA), Vienna, Austria
and
Department of eBusiness, School of Business, Economics and Statistics,
University of Vienna, Vienna, Austria
e-mail: natalia.kryvinska@univie.ac.at
http://www.sba-research.org/

Christine Strauss
Department of eBusiness, School of Business, Economics and Statistics,
University of Vienna, Vienna, Austria
e-mail: christine.strauss@univie.ac.at

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 167–187.
DOI: 10.1007/978-3-642-34952-2_7

This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability" [45, 47].

- "A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business processes. Services are available to interact with these 'smart objects' over the Internet, query and change their state and any information associated with them, taking into account security and privacy issues" [46, 47].
- "In the future the Internet of Things may be a non-deterministic and open network in which auto-organized or intelligent entities (Web services, SOA components), virtual objects (avatars) will be interoperable and able to act independently (pursuing their own objectives or shared ones) depending on the context, circumstances or environments" [47, 48].

Besides, the emergence of the virtual enterprises and inter-enterprise virtual platforms is the opportunity to enable productivity gains as well as flexibility and responsiveness to customer and market dynamics that enterprises need to be competitive in today's environment. But, to take advantage of this opportunity and to succeed in this new environment, enterprises need to create service delivery and communication strategies that establish tighter connections among their employees as well as with partners and suppliers. Central to this focus is the service architecture that powers the enterprise interactions with customers, between enterprises on this platform, as well as the processes for delivering value to customers and shareholders.

To enable enterprises to implement business strategies that are truly driven by a customer focus, it is required:

- create an instant and seamless connection across enterprises: linking people, processes, systems and networks so the customers are better served;
- provide seamless access to critical communications and business information to facilitate better, faster decisions and enable a more competitive enterprise;
- deliver personalized services needed to build long-term customer relationships.

Today's business environment can be seen as a logical effect of eBusiness and advanced integrated networks, including the IoT, that have transformed business processes. The new "Net" and its applications deployment models have revolutionized the way businesses interact, collaborate and transact business with customers, suppliers, partners, employees and shareholders [1 ÷ 7, 51].

Thus, the purpose of our work is to develop a model enabling enterprises to attach their customers more tightly by an effective service delivery enabling business processes at every step of the way.

We classify here the service delivery platforms into the categories: Web 2.0, Service-Oriented Architecture (SOA) and Virtualization of Services. Sections 2 and 3 respectively discuss contemporary concepts, architectures and technologies to support an effective services delivery and enable high services availability on SDP. Section 4 presents our approach towards hypothetical modeling of the service availability in the SDP environment. The numerical patterns/shapes of the conceptual model are presented in the section 5. Some practical model cases from the research literature are analyzed in the section 6. Section 7 concludes the chapter and identifies future research.

2 Contemporary Concepts, Architectures and Technologies to Support Effective IoT-enabled Business Services Delivery

In relation to the definitions and characterizations exposed into the Introduction, the SDP (Service Delivery Platform) bridges distributed inter- and intra-enterprise IT environments over communications wireline/wireless networks to streamline new-services deployment and delivery. By combining technologies that deliver services to end users, SDPs facilitate communication between OSS/BSS (Operations Support System/Business Support System), applications spanning heterogeneous computing platforms as well as interfaces with physical network elements [8 ÷ 12, 49, 51].

Thus, an effective SDP must handle high-volume traffic loads with carrier-grade reliability, and support a dynamic mix of service offerings to a growing subscribers' base over constantly changing network configurations. SDPs need to include powerful quality assurance and performance monitoring tools for quick rollout and high quality of new services, fulfillment of partner SLAs (Service Level Agreements), problems preemption and rapid decision for both IT and Business Operations departments [8 ÷ 12].

Besides, the Web 2.0, Service-oriented Architecture (SOA), and Cloud Computing [14, 49, 50] are the most discussed issues among IT architects and business executives. Both technologies are on the edge of an exponential growth over the next few years, due to their flexibility, cost effectiveness, and ease of integration. Each technology creates highly distributed composite applications (e.g., other words - mash-ups) that connect components or subsystems to form higher-level functional systems or target applications, and meet the following requirements:

- robust reliability with minimized latency and high availability;
- multiple layers of security to protect against general and XML-specific attacks;
- off-load of resource-intensive functions;
- XML acceleration for faster and more efficient performance;
- consistent high quality services;
- highly productive, innovative composite applications that combine business applications with communication and collaboration services.

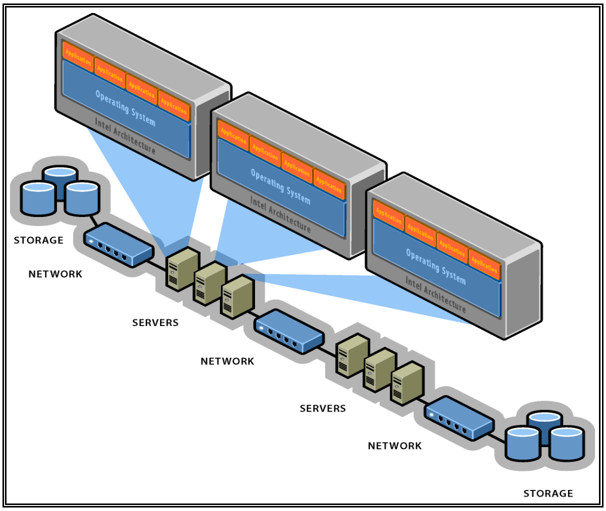


Fig. 1 Complex networked IT Architecture [13]

The growing challenge for enterprise IT architects is that because the composite applications are highly distributed, interactions between components may require several traversals across various areas of the network, each increasing the possibility of inconsistent performance or security problems. And, in turn, it decreases the availability of business services or processes built upon these composite applications. This issue becomes even more critical when Web 2.0 technologies are applied in order to leverage resources outside of the enterprise domains, using external networks and the Internet. To assure that all elements of broaden composite applications operate quickly, efficiently, and securely, a pervasive, reliable networked architecture is required. Besides, it is also important to understand the additional role the underlying infrastructure plays in Web 2.0 and SOA applications [14 ÷ 18, 20].

However, the reason why enterprises have not yet applied high-availability distributed SDPs to their inter- and intra- infrastructures - is due to three primary challenges (Fig. 1):

1. costs to implement additional hardware can be quite excessive, also including expenses associated with additional tools and training;
2. complexity of developing and managing SDP solutions may be overwhelming for some organizations, including lock-step hardware and software upgrades;
3. reliability can be questionable due to limited testing possibilities of the complex solutions deployed [13, 15 ÷ 18].

The Web 2.0 mashups, e.g., web applications that combine data from more than one source into a single integrated tool, and the SOA are the most illustrative models of what today are generically known as composite applications. Composite applications (e.g., building blocks or milestones of inter-enterprise SDPs) are application systems that are fundamentally enabled by network connectivity.

They are composed of loosely coupled subsystems to form a higher-level functional system or target application. These subsystems can be data sources or services that perform a particular function, accepting input from and providing output to the target application. Composite applications can provide tremendous flexibility and, properly designed, offer high levels of business agility and productivity due to their ability to be reconfigured relatively quickly. The underlying model of composite design is similar in both Web 2.0 and SOA applications (Fig. 2).

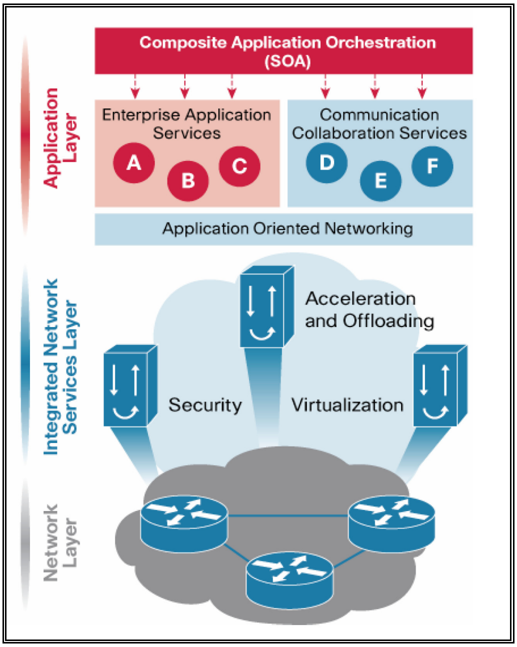

Composite Application Orchestration (SOA)
Application Layer
Enterprise Application Services
A
B
C
Communication Collaboration Services
D
E
F
Application Oriented Networking
Integrated Network Services Layer
Acceleration and Offloading
Security
Virtualization
Network Layer

Fig. 2 SDP model built upon SOA, Web 2.0 and other modern technologies [14]

In addition, the Web 2.0 is an evolutionary phenomenon that can be viewed from different perspectives. The user perspective encompasses a powerful trend toward user empowerment: Web 2.0 environments are greatly enriched by the simple premise that users should also be allowed to be content providers. The most influential examples of the synergies sets in this approach include wikis (Wikipedia), popular blogs (Engadget), photo-sharing sites (Flickr), video-sharing sites (YouTube), and social networks (MySpace, Facebook). The two main elements of this prosperous delivery environment are the concept of software as a service (SaaS) and mashup applications. SaaS allows the Web browser to challenge traditional desktop software when it comes to application delivery. Mashup applications unite data from different sources using open, intuitive protocols such as Extensible Markup Language (XML) and Representational State Transfer (REST) to create a contextually relevant presentation. By presenting data in innovative ways, mashups can significantly boost productivity, breaking down artificial barriers in data interpretation. A well-known example of a mashup is to combine data from a Web application such as Google Maps that uses a REST interface, an architectural model for designing the easy flow of information via the Web.

On the other hand, SOA is a design and architecture paradigm centered on the creation of component ‘services’ that can be combined to create business application systems. SOA is an architectural philosophy that does not specifically require or align itself with any particular technology set. It is focused on providing a tighter affiliation between business process and IT architecture in a modular fashion, with the goals of providing business agility, flexibility, and cost-effectiveness in long-term use. The SOA service components exhibit some typical core characteristics that deliver on the promise of flexibility, ease of integration, and cost benefits. They are:

- loosely coupled using defined interfaces;
- internal functions, structure, and states are completely internalized and irrelevant to other components in the system;
- can be combined and recombined as needed;
- discoverable by other existing or new components or systems within the architecture;
- amenable to service agreements, e.g., capable of providing and adhering to publishable service definitions that outline functional capabilities, interfaces, inputs, and outputs.

The general idea of service orientation is to decompose functional processes into modular services or sub-processes that can be served by IT systems to optimally support higher-level business processes. The web services protocols such as XML and Simple Object Access Protocol (SOAP) currently serve as the standard technology set for SOA [14 ÷ 17, 19, 20, 51].

In the next sections we examine and model how services availability on enterprise SDPs can be increased when applying mathematical methods in combination with different available technologies.

3 Enabling High Services Availability on SDP

The enterprises with large-scale IT infrastructures are facing a double-edged challenge. The financial pressures exerted on IT budgets by the never-ending increase in demand for storage and compliance requirements, along with the ever-present need to provide resilient business continuity solutions.

The advent of SOA has led to an unpredictability of demand. The assumption from end users, more so now than ever, and the applications demanded to be available 24x7, 365 days a year. This need for trust in systems is an essential part of expectations from customers, partners and employees.

The benefits of virtualization in being able to reduce costs for large-scale organizations are undeniable. However, while server virtualization has brought major benefits, it can also introduce potential vulnerabilities. In a physical server environment, loss of a single server has significantly less impact than in the virtual world where, workload dependent, the consolidation ratio of virtual machines running on a single physical server could be in the 10-15x range. A physical server failure can affect all of the virtual machines and applications running on that piece of hardware. Similarly failure of the virtualization layer itself impacts all running virtual environments. The complexity of this scenario grows as organizations standardize on server virtualization and deploy tier one applications in a virtual server environment. In short virtualization, while hugely effective in what it does, is not enough on its own to provide safeguards against unplanned downtime. Furthermore, while server virtualization can address consolidation at the server level, it can be found desirable at the level of storage, data and applications.

Some enterprises may not be in a position to deploy a grid infrastructure. The reasons for this may be one of enterprise size, footprint size, IT policy, outsourcing, lack of budget. In these circumstances it is generally recognized as good practice for applications with non-intensive workloads to use server virtualization in order to maximize consolidation.

However, where maximizing consolidation, availability and agility are of paramount importance, a combination of server virtualization and grid-based solutions are the best way to maximize the benefits of consolidation, availability and agility. Working in tandem, they can ensure enhanced server virtualization, the ability to dynamically scale within and across nodes, and the dynamic resizing of virtual nodes.

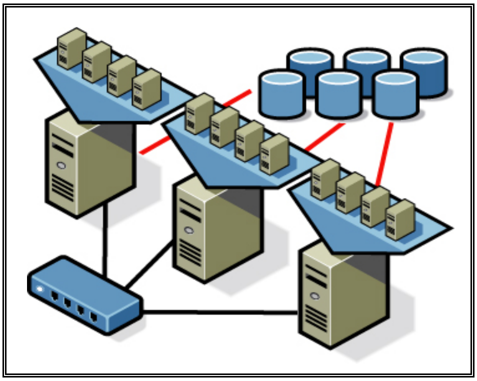


Fig. 3 Schematic arrangement of server virtualization with other technologies shaping SDP [13]

Compared to other models of computing, IT systems designed and implemented in the grid style deliver a higher quality of service, at a lower cost, with greater flexibility. Higher quality of service results from having no single point of failure, a powerful security infrastructure, and centralized, policy-driven management.

Lower costs, meanwhile, derive from increasing the utilization of resources and dramatically reducing management and maintenance costs. Rather than dedicating a stack of software and hardware to a specific task, all resources are pooled and allocated on demand, which eliminates underutilized capacity and redundant capabilities. Grid and cloud computing also enable the use of smaller individual hardware components, which reduces the cost of each individual component and provides more flexibility to devote resources in accordance with changing needs.

The progressive enterprises have to implement a combination of server virtualization with grid and cloud computing to take advantage of database consolidation, running multiple, disparate workloads on the shared resources of the grid. The consequence is a more available, scalable, flexible and cost effective infrastructure resulting in better service levels to customers, users and partners (see an example in Fig. 3).

Latest cloud-based solutions are now available that also offer all the benefits of server virtualization to single-instance databases on a physical hardware infrastructure. Many databases can be consolidated into a single cluster with minimal overhead while providing the high availability benefits of failover protection, online rolling patch application, as well as rolling upgrades for the operating system.

With these next-generation solutions, the limits of server scalability decrease and if applications grow to require more resources than a single node can supply, they can be easily upgraded online. If the node becomes overloaded, users can migrate the instance to another node in the cluster using an online migration utility with no downtime for application users [21 ÷ 25, 49, 50, 51].

4 Hypothetical Model to Increase Service Availability on Distributed SDPs

The examination and improvements-modeling of system performance issues are essential tools in the development and engineering processes that can be used at all stages of the lifecycle of business services (Fig. 4) [18, 27].

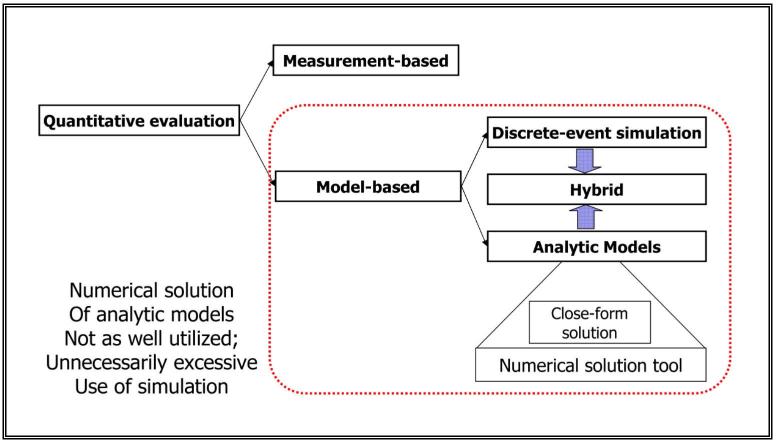


Fig. 4 Theoretical Modeling Procedure [26]

Therefore, on the primary (i.e., pre-prototyping) stage, in order to model the SDP behavior as a reaction to the services availability variation, we consider it as a network of queues (M/M/2/K/K model) where the total number of customers (e.g., enterprise servers) is fixed and limited since no customers are allowed to arrive or depart. This network is called closed network, which can be analyzed using Markov chains. And, the steady-state occupancy distribution has a product form under assumptions similar to those used for open networks.

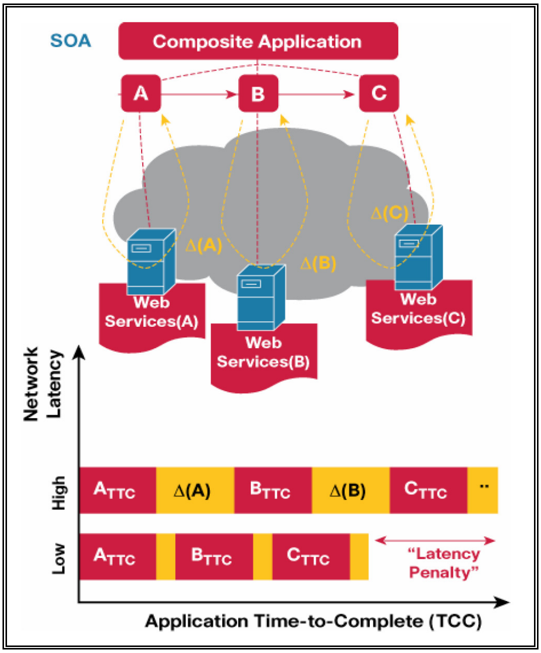


Fig. 5 Large scale, distributed SDP [14]

Traditional services were provided by the service logic and data resident within the local machine. The capacity for these services is very much determined by the architecture and component capacities within the service node. The SDP has a distributed architecture in which service logic is executed cooperatively by different network elements that can be geographically dispersed (Fig. 5).

An M/M/2/K/K is well known as the machine repair model or the cyclic queue model. In its context, there are *K* jobs cycling in a system consisting of *K* terminals or, in our case enterprise sub-servers, and two central Servers with a work queues. A job (e.g., request for a business service) is sent from a user/workstation/sub-server to the Server after an exponentially distributed "think time" and after being processed by the Server the job enters another think phase at a user side. The input and output messages of a transaction are treated as a single composite service. Also, the "think" time and the Server processing time are considered as an average "operating" time.

In general, an M/M/m/K/K model is presented a model of a system with K users and m parallel servers. There are at least as many users as servers. If $K < m$, then m - K servers are never used and may be discarded. The user' think times are distributed exponentially with parameter λ. Service times at all Servers are distributed exponentially with parameter μ. The system is in state j (j = 0, 1,…, K) if j users are waiting for their requests to be completed and K - j users are thinking. The instantaneous transition rate from state j to state j + 1 is equal to [27 ÷ 33, 51]:

$$\lambda_j = (K - j)\lambda, \qquad j = 0, 1, ..., K - 1; \tag{1}$$

since each of the thinking users submits requests at rate λ. The rates from state j to state j-1 depend on whether the number of requests is less than the number of Servers, in a similar way to M/M/m/K/K model:

$$\mu_j = \begin{cases} j\mu & \text{for} \quad j = 1, 2, ..., m - 1 \\ m\mu & \text{for} \quad j = m, m + 1, K \end{cases} \tag{2}$$

The balance and normalizing equations yield

$$\begin{aligned} p_j &= \frac{K!}{(K - j)!\, j!} \rho^j p_0; \qquad j = 0, 1, ..., m - 1 \\ p_j &= \frac{K!}{(K - j)!\, m!\, m^{j-n}} \rho^j p_0; \quad j = m, m + 1, ..., K \end{aligned} \tag{3}$$

with ρ – server utilization and p_0 given by:

$$p_0 = \left[\sum_{j=0}^{m-1} \frac{K!}{(K - j)!\, j!} \rho^j + \sum_{j=m}^{K} \frac{K!}{(K - j)!\, m!\, m^{j-m}} \rho^j \right]^{-1} \tag{4}$$

The throughput, T, can be obtained either as the average number of requests completions, or as the average number of requests submissions, per unit time. The former approach requires the average number of busy Servers, r:

$$r = \sum_{j=1}^{m-1} j p_j + m \sum_{j=m}^{K} p_j \tag{5}$$

The expression for the throughput is then $T = r\mu$. Alternatively, we could find the average number of requests in service or in the queue, L:

$$L = \sum_{j=1}^{K} j p_j \tag{6}$$

Then the average number of thinking users is *K-L*. Since each of them submits requests at rate λ, the throughput is equal to $T = (K\text{-}L)\lambda$.

In the two special cases when $m = 1$ and $m = K$, the expressions have a simpler form. If there is a single Server, the steady-state probabilities are:

$$p_j = \frac{\rho^j}{(K-j)!}\left[\sum_{i=0}^{K} \frac{\rho^i}{(K-i)!}\right]^{-1}; \quad j = 0, 1, \ldots, K, \tag{7}$$

and, the throughput is equal to:

$$T = (1 - p_0)\mu. \tag{8}$$

When the number of Servers is equal to the number of users, no request has to queue and users do not interfere with each other in any way. The steady-state distribution of the number of requests in service is binomial [27 ÷ 33].

The average number of busy Servers is:

$$r = \frac{K\rho}{1+\rho}. \tag{9}$$

The throughput is given by [32, 33]:

$$T = \frac{K\lambda}{1+\rho} \tag{10}$$

5 Numerical Patterns of the Model

The conceptual model is a crucial tool in the development and engineering processes that can be used at all stages of the lifecycle of business services.

Besides, simple, approximate models have a high value in the early stages to uncover major performance problems, which affect the design of the architecture before the cost of an alteration is too high. The design or conceptual models support prompt prototyping, allowing researchers to go through the three important stages: predict, design, and comparison [18, 27].

Thus, the model presented in this chapter, has to be viewed as a concept/vision, based on classical queuing theory formulations and patterns. In this section, we figured the patterns for the certain input parameters of our conceptual model.

Accordingly, the results of the formula' calculations from the previous section have to be considered as the stencils or templates or prototypes, and certainly not as pragmatic values.

Accordingly, Table 1 includes numerical patterns of the probability of 0 customers in system for $m = 2$ servers. In Fig. 6, it is shown comparison of the probabilities that there are 0 customers in system ($K = 20$) for $m = 1$ and 2 servers. And, as it was expected, the probability that there are no customers in system, of course, is higher for $m = 2$.

Table 1 The Probability of 0 Customers in System ($M = 2$)

$\lambda\,(\mu=1)$	$p_0\,(K=1)$	$p_0\,(K=2)$	$p_0\,(K=5)$	$p_0\,(K=10)$	$p_0\,(K=20)$
0.001	0.999001	0.998003	0.995015	0.990055	0.980208
0.1	0.909091	0.826446	0.618592	0.367955	0.086302
0.2	0.833333	0.694444	0.392711	0.120186	$9.35399 \ast 10^{-4}$
0.3	0.769231	0.591716	0.253503	0.0340509	$7.86557 \ast 10^{-6}$
0.4	0.714286	0.510204	0.166091	$9.27757 \ast 10^{-3}$	$1.32061 \ast 10^{-7}$
0.5	0.666667	0.444444	0.110535	$2.66084 \ast 10^{-3}$	$4.13873 \ast 10^{-9}$
0.6	0.625000	0.390625	0.07481	$8.33697 \ast 10^{-4}$	$2.1027 \ast 10^{-10}$
0.7	0.588235	0.346021	0.0515277	$2.87008 \ast 10^{-4}$	$1.5511 \ast 10^{-11}$
0.8	0.555556	0.308642	0.0361251	$1.07881 \ast 10^{-4}$	$1.5343 \ast 10^{-12}$
0.9	0.526316	0.277008	0.0257698	$4.38542 \ast 10^{-5}$	$1.9209 \ast 10^{-13}$
0.999	0.500250	0.25025	0.0187502	$1.92489 \ast 10^{-5}$	$2.9695 \ast 10^{-14}$

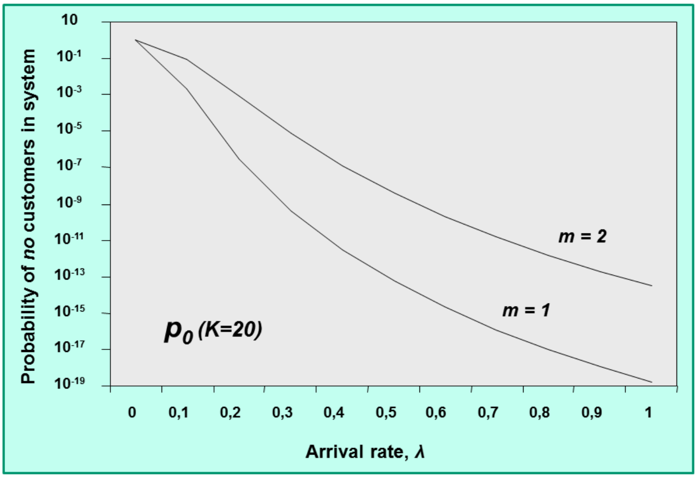


Fig. 6 The probability of no customers in system ($K = 20$) for $m = 1$ and 2 servers

In Table 2 and Fig. 7 are given numerical forms of the probabilities for finding k (K = 1-20, m = 2) in system. In comparison with m = 1 Server system, the probability of k customers in system, when m = 2 Servers is decreasing, because service rate is growing.

Table 2 The Probability of K Customers in System (K = 2; M = 2)

$\lambda\,(\mu=1)$	*p (K=1)*	*p (K=2)*	*p (K=5)*	*p (K=10)*	*p (K=20)*
0.001	0.000999	0.001996	0.0049751	0.00990055	0.019604158
0.1	0.0909091	0.165289	0.309296	0.367955	0.172604712
0.2	0.166667	0.277778	0.392711	0.240372	0.003741596
0.3	0.230769	0.35503	0.380255	0.102153	$4.7193*10^{-5}$
0.4	0.285714	0.408163	0.332182	0.0371103	$1.0565*10^{-6}$
0.5	0.333333	0.444444	0.276339	0.0133042	$4.1387*10^{-8}$
0.6	0.375	0.46875	0.22443	0.00500218	$2.5232*10^{-9}$
0.7	0.411765	0.484429	0.180347	0.00200906	$2.172*10^{-10}$
0.8	0.444444	0.493827	0.1445	0.00086305	$2.4549*10^{-11}$
0.9	0.473684	0.498615	0.115964	0.00039469	$3.4576*10^{-12}$
0.999	0.49975	0.5	0.0936573	0.0001923	$5.933*10^{-13}$

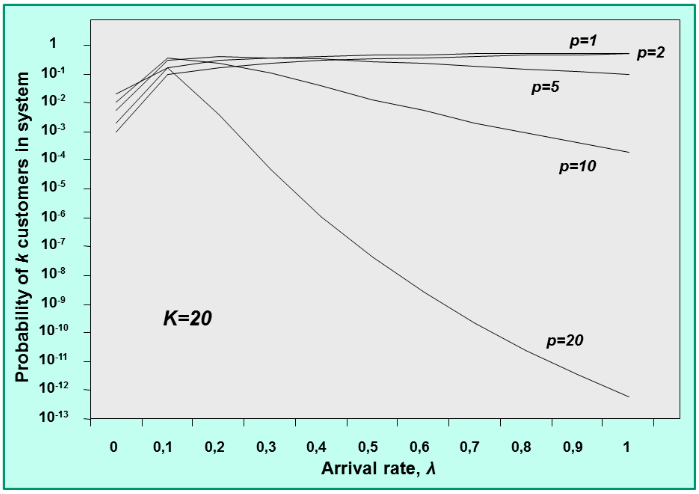


Fig. 7 The probability of k customers in system (K = 20; m = 2)

The average number of customers in system (e.g., in the queue or service) L is presented in Table 3 (e.g., some numerical shapes) and plotted in Fig. 8.

Table 3 Average Number of Customers in System (M=2)

λ (μ=1)	*L (K=1)*	*L (K=2)*	*L (K=5)*	*L (K=10)*	*L (K=20)*
0.001	0.000999	0.004991	0.00797	0.0178706	0.520955
0.1	0.0909091	0.421487	0.565494	0.933449	9.09961
0.2	0.166667	0.722223	0.837156	1.07753	5.16433
0.3	0.230769	0.940829	0.966054	1.06821	3.86458
0.4	0.285714	1.10204	1.02606	1.06317	3.13407
0.5	0.333333	1.22222	1.05412	1.06742	2.73696
0.6	0.375	1.3125	1.06818	1.07318	2.48467
0.7	0.411765	1.38062	1.07654	1.07855	2.30245
0.8	0.444444	1.4321	1.08277	1.08363	2.16323
0.9	0.473684	1.47091	1.08826	1.08866	2.05468
0.999	0.49975	1.49975	1.09341	1.0936	1.96996

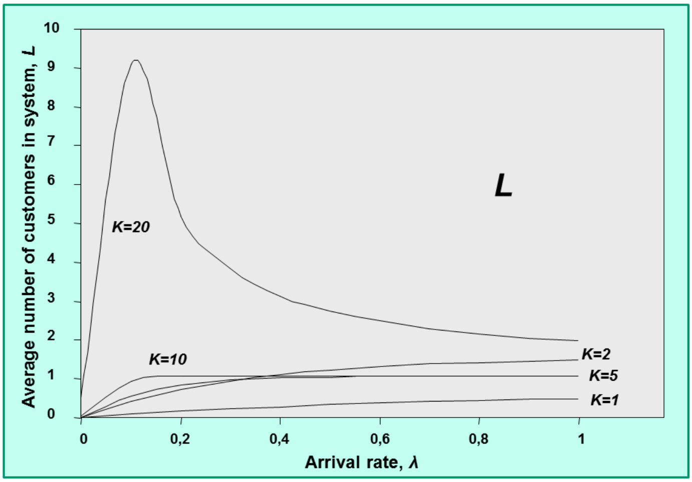


Fig. 8 Average number of customers in system (*m* = 2)

In Table 4 and Fig. 9 we show the average time in system *T* (e.g., throughput), when *m* = 2 servers, and distribution of the service time is exponential.

6 Literature Case Studies – Business Services Availability in the Collaborative Mobile Environment

As it is observed and described correctly in [34], Web services and mobile data services are the newest trends in information systems engineering in wired and

Table 4 The Average Time in System (e.g., Throughput) T ($M = 2$)

λ (μ=1)	*T (K=1)*	*T (K=2)*	*T (K=5)*	*T (K=10)*	*T (K=20)*
0.001	0.000999	0.001995	0.004992	0.00998213	0.019479
0.1	0.0909091	0.157851	0.443451	0.906655	1.09004
0.2	0.166667	0.255555	0.832569	1.78449	2.96713
0.3	0.230769	0.317751	1.21018	2.67954	4.84063
0.4	0.285714	0.359184	1.58958	3.57473	6.74637
0.5	0.333334	0.38889	1.97294	4.46629	8.63152
0.6	0.375	0.4125	2.35909	5.35609	10.5092
0.7	0.411765	0.433566	2.74642	6.24501	12.3883
0.8	0.444445	0.45432	3.13378	7.1331	14.2694
0.9	0.473684	0.476181	3.52056	8.02021	16.1508
0.999	0.49975	0.49975	3.90269	8.89749	18.012

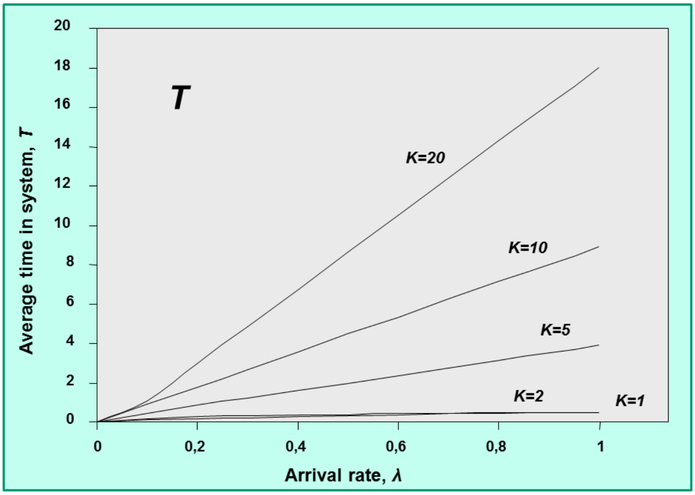


Fig. 9 The average time in system (e.g., throughput) T ($m = 2$)

wireless domains, respectively. Web services have a broad range of service distributions while mobile phones have large and expanding user base. Further, Srirama et al. [34] comment extensively that current generation of mobile devices like smart phones, PDAs and other consumer devices, the wireless market is expanding very fast. People are using such high-end mobile phones and devices for wide range of applications like mobile banking, location based services, e-learning etc. The situation also brings out a large scope and demand for software applications for such high-end mobile devices. The biggest advantage of Web Services lies in its simplicity in expression, communication and servicing [34].

On the other hand, Tan et al. [35] discovered appropriately that adapting present Web services to the mobile environment has raised new Quality of Service (QoS) challenges for both service provider and requester. In a mobile Web services (MWS) scenario, a successful service invocation will depend on service application, condition of mobile network, and capabilities of mobile device. Hence, from a client perspective, the availability of a service will depend on all three factors.

They [35] also develop an Availability Checking Model (ACM), to determine the availability status of MWS by processing and referring to end-to-end service QoS parameters. This model introduces availability metrics to quantify service availability checking points.

Other paper from Gebaur et al. [36] claims that the deployment of Web service in mobile environment will contribute to the success of mobile commerce.

In the [37] authors study how to sustain the high availability and quality of Web services by using communities of Web services. They define that availability is the probability that a Web service is in functioning condition at a specific time. Communities of Web services gather Web services that provide the same functionality, but not necessary with the same quality. Within this framework, an available Web service can be selected or even substituted by another one when it fails. In this research, authors investigate also how to manage the community to sustain not only the high availability but also to guarantee the quality of service expected by the user.

Besides, authors [37] explore and describe accurately that Web services are intensively used for developing loosely-coupled, inter-enterprise business processes. Ideally, a Web service is required to accept all the requests of the users. However, a Web service can get busy with an overloaded demand and thus, cannot offer a good quality of service to the number of users that exceeds its capacity. This situation would cause inefficiency in offering service quality and consequently lead to a drop in the satisfaction value of the Web service that is assigned by the users. Thus, sustaining high availability of a Web service is a challenging but crucial issue. For a Web service, failing to respond with an acceptable quality of service would cause a negative satisfaction of the users, and hence, drop the future number of requests [37].

In [38], the authors present a tool that bears the service quality within workflow management systems. They use mathematical models to meet specific goals for performance and availability due to outages of individual servers [37, 38].

As it is stated in the next [39] paper, for Web services to become a universal communication paradigm, mobile devices enabled with Web services should be considered as an equal participant of the service-oriented architecture. At this point, mobile devices play the role of clients, providers, or even brokers. To establish a distributed application framework, the paper [39] presents a light-weight framework for hosting Web services on mobile devices. The proposed framework contains several built-in functionalities such as the processing of SOAP messages, the execution and migration of services, the management of context and service directory, and the publishing and discovery of services.

The authors [39] indicate also that it is difficult to apply Web services to a mobile environment because most of existing Web services have been developed targeting desktops and wired environments. However, a mobile device will be changed into a powerful server, which can provide an independent service. Furthermore, it is difficult to provide Web services on a mobile environment because of the intermittent disconnection of wireless network and the frequent change of context information. For example, if a Web service provider disappears or changes its position, then location information becomes invalid. If clients are not notified about these changes and their requests are not adjusted, the whole process is brought to a halt. However, if a Web service would be migrated to a relevant device, a user could receive the required service seamlessly. In this case, the migration of Web services is an essential technique for providing a user with services on a mobile environment [39].

In the next paper [40], the authors discuss the integration of QoS-aware search and personalization algorithms in order to discover effectual Web Services for the case of mobile web users. In recent years, mobile search services have gained a lot of interest in the mobile communication markets. A major drawback for mobile applications is the fact that mobile devices cannot process complex business logic due to their limited capabilities [40].

The ability to access and consume web services over mobile devices would allow users to expand the capabilities of their device by providing the necessary processing power [40, 41].

Awareness of non-functional features such as performance, reliability, security, and cost would enhance the automated WS selection and composition process. In the case of mobile Web Services, the need for qualitative search is extremely crucial since mobile environments bear additional constrains like limited bandwidth, unpredictable response time and high probability of packet loss [35, 40].

And to conclude, the performance and quality of a web service are important factors and are expected by parties utilizing the service, as mentioned in [42]. The authors propose and deploy in this paper a cost efficiency solution to provide for web services in a unique set of situation. They also claim that Web services present a simple method for creating interoperable applications directed towards the distributed environment. With this paradigm, distributed applications can be implemented through the composition of multiple web service components that are offered by a variety of vendors residing in various geographical locations. Since web services are generally part of an application composition, the preservation of quality and performance of individual services at some efficient level, are essential to limit the possibilities of partial failure and enhance overall application responsiveness [42, 43, 44].

7 Conclusions

We have examined in this chapter how IoT, Web 2.0, SOA as well as other modern concepts and technologies (like cloud computing) along with composite applications, integrated network services, and newest underlying technologies (like servers virtualization in combination with grid, cloud computing and consolidating techniques) can help to create an optimal SDP foundation enabling to fulfill the potential as the next “disruptive force” of innovation on eBusiness platforms. We have also discussed and modeled here how SDP can provide key categories of critically important business services, e.g., basic enablement, performance optimization with application enrichment, to ensure services availability, reliability, scalability, security, and predictable performance across diverse network/IT environments, and to support optimal alignment of composite applications with the business process they support [14 ÷ 17, 19, 20, 49, 50].

Besides, the analysis and modeling of system performance issues are essential tools in the development and engineering processes that can be used at all stages of the lifecycle of business services. Simple, approximate models have a high value in the early stages to uncover major performance problems, which affect the design of the architecture before the cost of rectification is too high. The design tools support rapid prototyping, allowing users to go through the three important stages: predict, design, and comparison. The questions of the development of new modeling methods for rapid analysis, and some others, like new performance standards, and closer connections between performance analysis and service design are the most interesting for the future wide deployment of business services [18, 27, 51].

References

1. White paper. Avaya Communication Architecture. LB1842, Avaya (March 2006), `http://www.avaya.com`
2. Davidow, W.H., Malone, M.S.: The virtual corporation. HarperCollins Publishers, New York (1992)
3. Jägers, H.P.M., Jansen, W., Steenbakkers, G.C.A.: Characteristics of Virtual Organizations. In: Proceedings of the VoNet Workshop, Bern, April 27-28 (1998)
4. Lippis, N.: Web Services & SOA Redefine IP Telephony Landscape. The Lippis Report (51) (February 2006)
5. Venkatraman, N., Henderson, J.C.: Real strategies for virtual organizing. Sloan Management Review (Fall 1998)
6. Johnson, E.: Elusive Integration: Linking Sales and Operations Planning. ASCET 7 (2005)
7. Palmer, J.W., Speier, C.: A typology of virtual organizations: an empirical study. In: Proceedings of the Association for Information Systems 1997 Americas Conference, Indianapolis, August 15-17 (1997)
8. Solution Brief, BEA Service Delivery Platform with Wily Performance Monitoring Enabler for Telecommunications. PSB1433E0906-1A, BEA Systems, a subsidiary of Oracle Corporation (September 2006)
9. Weerawarana, S., Curbera, F., Leymann, F., Storey, T., Ferguson, D.F.: Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall (2005)

10. Cho, J.-H., Yu, B.-F., Lee, J.-O.: Deploying Application Services Using Service Delivery Platform (SDP). In: Ata, S., Hong, C.S. (eds.) APNOMS 2007. LNCS, vol. 4773, pp. 575–578. Springer, Heidelberg (2007)
11. Christian, R., Hanrahan, H.: Defining Generic Architectural Requirements for the Service Delivery Platform. In: Proceedings of the Southern African Telecommunications and Networks Conference (SATNAC 2007), Mauritius (September 2007)
12. Demydov, I., Kryvinska, N., Strauss, C., Klymash, M., Ivanochko, I.: Enterprise Distributed Service Platforms - an Approach to the Architecture and Topology. In: Proceedings of the Emerging Research and Projects Applications Symposium, ERPAS, in conjunction with MoMM 2009, December 14-16, Kuala Lumpur, Malaysia (2009)
13. On-line Presentation, VMware (2004), `http://www.synegi.com/docs/vmware-synegi-presentation.ppt`
14. White Paper, Cisco Service-Oriented Network Architecture: Support and Optimize SOA and Web 2.0 Applications. Cisco Systems, C11-450967-00 1/08 (2008)
15. Kryvinska, N., Auer, L., Strauss, C.: Managing an Increased Service Heterogeneity in Converged Enterprise Infrastructure with SOA. International Journal of Web and Grid Services (IJWGS) 4(4) (2008)
16. Christian, R., Hanrahan, H.E.: Defining a Service Delivery Platform Architecture using Generic IMS and SOA Concepts. In: Proceedings of 11th International Conference on Intelligence in Service Delivery Networks (ICIN 2007), Bordeaux (October 2007)
17. Radhakrishnan, V., et al.: PIAF: An Application Framework for Unlocking IMS Engendered Network Capabilities. In: Programming Models for the IMS, 10th International Conference on Intelligence in Service Delivery Networks (ICIN), Bordeaux, France (May 2006)
18. Kryvinska, N.: Converged Network Service Architecture: A Platform for Integrated Services Delivery and Interworking. Electronic Business series, vol. 2. International Academic Publishers, Peter Lang Publishing Group (2010)
19. Kryvinska, N., Auer, L., Strauss, C.: The Place and Value of SOA in Building 2.0-Generation Enterprise Unified vs. Ubiquitous Communication and Collaboration Platform. In: Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies (UBICOMM 2009), NexTech 2009, Sliema, Malta, October 11-16 (2009)
20. Demydov, I., Kryvinska, N., Klymash, M.: An Approach to the Flexible Information/Service Workflow Managing in Distributed Networked Architectures. In: Proceedings of the International Workshop on Design, Optimization and Management of Heterogeneous Networked Systems (DOM-HetNetS 2009), in conjunction with ICPP 2009, Vienna, Austria, September 22-25 (2009)
21. White paper, Making the Business Case for Data Centre Consolidation. Oracle and IDG Global Solutions (2009), `http://www.oracle.com/technology/tech/grid/index.html`
22. Kryvinska, N., Auer, L., Strauss, C.: SOI Framework for the Efficient Management of Complex Resource-Intensive Applications on Constrained Devices. In: Proceedings of the International Workshop on Design, Optimization and Management of Heterogeneous Networked Systems (DOM-HetNetS 2009), in conjunction with ICPP 2009, Vienna, Austria, September 22-25 (2009)
23. Thein, T., Park, J.S.: Availability analysis of application servers using software rejuvenation and virtualization. Journal of Computer Science and Technology 24(2) (March 2009)

24. Alonso, J., Silva, L., Andrzejak, A., Silva, P., Torres, J.: High-available grid services through the use of virtualized clustering. In: Proceedings of 8th IEEE/ACM International Conference on Grid Computing, Austin, Texas, September 19-21 (2007)
25. Goth, G.: Virtualization: Old Technology Offers Huge New Potential. IEEE Distributed Systems Online 8(2), 3 (2007)
26. Trivedi, K.: Dependability, Security and Survivability Models, Keynote. In: Proceedings of 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), Washington, D.C., USA, October 25-28 (2009)
27. Kryvinska, N., Nguyen, H.M.: Large Intelligent Network Modeling Using M/M/2/K/K System. In: Proceedings of IEEE 9th Asia Pacific Conference on Communications (APCC 2003), Penang, Malaysia, September 21-24 (2003)
28. Cardellini, V., Colajanni, M., Yu, P.S.: Dynamic Load Balancing on Web-server Systems. IEEE Internet Computing 3(3) (May/June 1999)
29. Emstad, P.J., Osland, P.-O.: Dynamic load balancing in a two-server system. In: Proceedings of the 17th International Teletraffic Congress (ITC-17), Salvador da Bahia, Brazil (September 2001)
30. Ackeley, R., Elvidge, A., Ingham, T., Shepherdson, J.: Network Intelligence – Performance by Design. IEICE Transactions on Communication E80-B(2) (February 1997)
31. Masuda, E., Mishima, T., Takaya, N., Nakai, K., Hirano, M.: A Large-Capacity Service Control Node Architecture Using Multicasting Access to Decentralized Databases in the Advanced Intelligent Network. IEICE Transactions on Communication E84-B(10) (October 2001)
32. Kleinrock, L., Gail, R.: Solution Manual for Queueing Systems. Volume II: Computer Applications. Technology Transfer Institute (1986)
33. Mitrani, I.: Probabilistic modeling. University Press, Cambridge (1998)
34. Srirama, S.N., Jarke, M., Prinz, W.: A Mediation Framework for Mobile Web Service Provisioning. In: Proceedings of Enterprise Distributed Object Computing Conference Workshops, IEEE International, 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW 2006), Hong Kong, China, October 16-20, p. 14 (2006)
35. Tan, K.-L., Mustapha, S.M.F.D.S.: Measuring Availability of Mobile Web Services. In: Proceedings of the 2006 International Conference on Semantic Web & Web Services (SWWS 2006), Las Vegas, Nevada, USA, June 26-29, pp. 137–142 (2006)
36. Gebauer, J., Shaw, M.: Success factors and impacts of mobile business applications: Results from a mobile eprocurement study. International Journal of Electronic Commerce, 19–41 (Spring 2004)
37. Lim, E., Thiran, P.: Sustaining High-Availability and Quality of Web Services. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 560–565. Springer, Heidelberg (2010)
38. Gillmann, M., Weikum, G., Konner, W.: Workflow management with service quality guarantees. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 228–239 (2002)
39. Kim, Y.-S., And Lee, K.-H.: A Light-weight Framework for Hosting Web Services on Mobile Devices. In: Proceedings of Fifth European Conference on Web Services (ECOWS 2007), pp. 255–263 (2007)
40. Sakkopoulos, E., Adamopoulou, P., Tsakalidis, A.K., Sioutas, S., Manolopoulos, Y.: Personalized selection of web services for mobile environments: the m-scroutz solution. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES 2009), Article 38. ACM, New York (2009)

41. Tergujeff, R., Haajanen, J., Leppänen, J., Toivonen, S.: Mobile SOA: Service Orientation on Lightweight Mobile Devices. In: Proceedings of IEEE International Conference on Web Services, USA, pp. 1224–1225 (2007)
42. Pratistha, D., Zaslavsky, A., Cuce, S., Dick, M.: Performance Based Cost Models for Improving Web Service Efficiency Through Dynamic Relocation. In: Bauknecht, K., Pröll, B., Werthner, H. (eds.) EC-Web 2005. LNCS, vol. 3590, pp. 248–257. Springer, Heidelberg (2005)
43. Burghart, T.: Distributed Computing Overview. Quoin Inc., Cambridge (1998)
44. Waldo, J., Geoff, W., Wollrath, A., Kendall, S.: A Note on Distributed Computing. Sun Microsystem. SMLI TR-94-29 (1994)
45. CASAGRAS (Coordination and support action for global RFID-related activities and standardization), The EU-funded project, A new vision for the Internet (December 23, 2008), http://cordis.europa.eu/search/index.cfm?fuseaction=news.document&N_RCN=30283
46. SAP, http://services.future-internet.eu/images/1/16/A4_Things_Haller.pdf
47. Wikipedia – IoT, Alternative definitions, http://en.wikipedia.org/wiki/Internet_of_Things
48. Guo, B.: Living with Internet of Things. In: The Emergence of Embedded Intelligence (CPSCom 2011) (2011), http://www.ayu.ics.keio.ac.jp/members/bingo/research/EI_CPSCom.pdf
49. Bessis, N., Asimakopoulou, E., French, T., Norrington, P., Xhafa, F.: The Big Picture, from Grids and Clouds to Crowds: A Data Collective Computational Intelligence Case Proposal for Managing Disasters. In: 1st International Workshop on Emerging Data Technologies for Collective Intelligence (EDTCI 2010), in conjunction with the 3PGCIC 2010, Fukuoka, Japan, November 4-6, pp. 351–356 (2010)
50. Bessis, N., Asimakopoulou, E., Xhafa, F.: A Next Generation Emerging Technologies Roadmap for enabling Collective Computational Intelligence in Disaster Management. International Journal of Space-Based and Situated Computing (IJSSC) 1(1), 76–85 (2011)
51. Kryvinska, N., Strauss, C., Younas, M., van Thanh, D.: A Generic Approach to the Services Delivery in Enterprise eBusiness Platforms. In: The 7th International Symposium on Web and Mobile Information Services (WAMIS 2011), in conjunction with IEEE AINA 2011, Biopolis, Singapore, March 22-25, pp. 843–848 (2011)

Organizational Control Reconfigurations for a Robust Smart Power Grid

Evangelos Pournaras, Mark Yao, Ron Ambrosio and Martijn Warnier

Abstract. Large-scale cyber-physical infrastructures, such as the Smart Power Grid, are envisioned as some of the core elements of the future Internet of Things. These critical infrastructures move more and more beyond centralized management and control by system operators and administrators. Overloading and failures in the Smart Power Grid threaten the matching of demand-supply especially when new emerging technologies are integrated such as micro-generation, renewable energy resources and electrical vehicles. The integration of these technologies in the Smart Power Grid make the concept of Internet of Things highly applicable in the energy domain. The introduction of automated and decentralized protection mechanisms requires embedded control elements that perform organizational reconfigurations themselves in a spatially distributed environment. The dynamic input and output binding between such control elements is an example of an organization reconfiguration that is traditionally managed offline during the design phase of a cyber-physical system. An introduced computational intelligence for the purpose of such organizational control reconfigurations requires the interoperation with the rest of the control logic during runtime. This book chapter illustrates a model that makes this interoperation possible: ALSOS-ICS, the Application-level Self-Organization Services for Internet-scale Control Systems. Four incremental protection levels for the robustness of the Smart Power Grid illustrate the requirements of organizational control reconfigurations and the applicability of ALSOS-ICS in this domain.

1 Introduction

Cyber-physical systems within the Internet of Things are built by physical and software elements of embedded control whose organization has traditionally been an

Evangelos Pournaras · Martijn Warnier
Delft University of Technology, Section Systems Engineering
e-mail: {e.pournaras,m.e.warnier}@tudelft.nl

Mark Yao · Ron Ambrosio
IBM Thomas J. Watson Research Center
e-mail: {markyao,rfa}@us.ibm.com

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 189–206.
DOI: 10.1007/978-3-642-34952-2_8

offline design aspect. Input and output (I/O) of control elements are bound manually to form the control loops of a control application [27]. This is a design phase that is usually not automated and occurs before system runtime [9]. However, the Internet of Things require online and automated organizational control reconfigurations as cyber-physical systems scale and their elements become more interconnected, distributed and interactive. Reconfigurations become an integral part of the computational intelligence that control elements should have. In this case, organizational control means that the feedback control loops formed by the I/O binding of the control elements are organized within a large-scale networked and distributed environment of Internet of Things. Network abstraction, fault-tolerance, latency, limited connectivity and shared resources are some of the challenges that need to be addressed [24]. Introducing and modeling dynamic reconfigurations of the organizational control in an Internet-scale control system is challenging. This is because the physical assets of a cyber-physical system interoperate based on I/O signals and feedback loops, whereas, distributed software elements are usually event-based, information-driven and are built by multiple layers of network abstraction. This modeling gap introduces various integration and interoperation issues that are identified in literature [15, 24]. Cyber-physical systems without organizational control reconfigurations cannot support the emergent application requirements of Internet of Things. Therefore, the modeling of organizational control reconfigurations is relevant and connects both of these related research areas [22]. This book chapter illustrates a modeling approach of organizational control reconfigurations in the application domain of the Smart Power Grid. This is a critical application domain for the Internet of Things as a large number of physical and software assets of the traditional electrical grid become more interconnected, intelligent and self-aware of socio-technical factors that mandate their design and operation [20].

The protection of the Smart Power Grid from overloading or failures is a critical requirement that involves various organizational control reconfigurations such as adjusting the load of a power line, switching the power flow to alternative distribution lines, turning on backup generators and restoring the system after a black-out. Traditionally, these reconfigurations are coordinated manually by experienced systems operators supported by usually centralized data acquisition information systems [12]. This approach is limited and cannot remain as a long-term solution in the future Smart Power Grid. The introduction of micro-generation, renewable energy resources and electrical vehicles are some examples that indicate the future challenge and complexity of matching demand and supply within a robust and dynamic Smart Power Grid. Therefore organizational control reconfigurations in an environment of Internet of Things need to be dynamic, automated and coordinated by the computational intelligence of control elements able to interoperate for this purpose.

The first contribution of this book chapter is to illustrate how the interoperation of the organizational control reconfigurations with the rest of the control application logic, e.g. the protection of the Smart Power Grid, is made possible by modeling the dynamic I/O binding of control elements as a control system within the Internet of Things. This means that a number of software embedded control elements, running as a control application, dynamically configure the I/O binding of a second control application that requires this dynamic I/O binding.

The second contribution of this book chapter is the illustration of four incremental protection levels for the robustness of the Smart Power Grid. These protection levels are actual application scenarios of organizational control configurations concerning the Internet of Things in the domain of the Smart Power Grid. The application scenarios show why control elements need to dynamically bind with each other on-the-fly and which protection measures these reconfigurations support. Within the domain of the Smart Power Grid, dynamic I/O binding is not introduced as an alternative protection mechanism such as the ones of [8], but as the means to make these mechanisms more applicable within dynamic environments of the Internet of Things.

This book chapter is organized as follows: Section 2 illustrates the model for organizational control reconfigurations: ALSOS-ICS, the Application-level Self-Organization Services for Internet-scale Control Systems. Section 3 outlines four incremental application scenarios for the robustness of the Smart Power Grid that require dynamic organizational control reconfigurations. Section 4 discusses the approach of ALSOS-ICS in the application domain of the Smart Power Grid and in the more general development life-cycle of distributed embedded control systems. Finally, Section 5 concludes this chapter and outlines future work.

2 Dynamic I/O Binding Reconfigurations

Two control elements are bound if there is at least one output from the first control element wired to the input of a second control element. A binding reconfiguration is defined by the rewiring of the input/output of a control element to a different output/input respectively of the same or of another control element. A dynamic binding reconfiguration means that this rewiring is automated during system operation (online) with a minimum or absent centralized intervention. Dynamic I/O binding reconfiguration requires system elements that should be capable to perform these reconfigurations and should be able to interoperate with the rest of system control logic. Existing solutions provide an offline [9], external [33] and/or centralized [10, 14, 23] configuration of I/O bindings.

Our earlier work [29] introduces a model for application-level reconfiguration of dynamic I/O binding in Internet-scale control systems that is referred to in this chapter as ALSOS-ICS, the *Application-Level Self-Organization Services in Internet-scale Control Systems*. This model introduces a new type of control application that provides reconfiguration services for dynamic I/O binding to other control applications. This type of control application is modeled as a control system, built by control elements that are able to interact with other control elements of the same API but implemented for a different application scope. This approach allows a higher interoperability and modularity between control applications and a higher flexibility, integration and applicability of dynamic I/O binding reconfigurations in the domain of cyber-physical systems and Internet of Things.

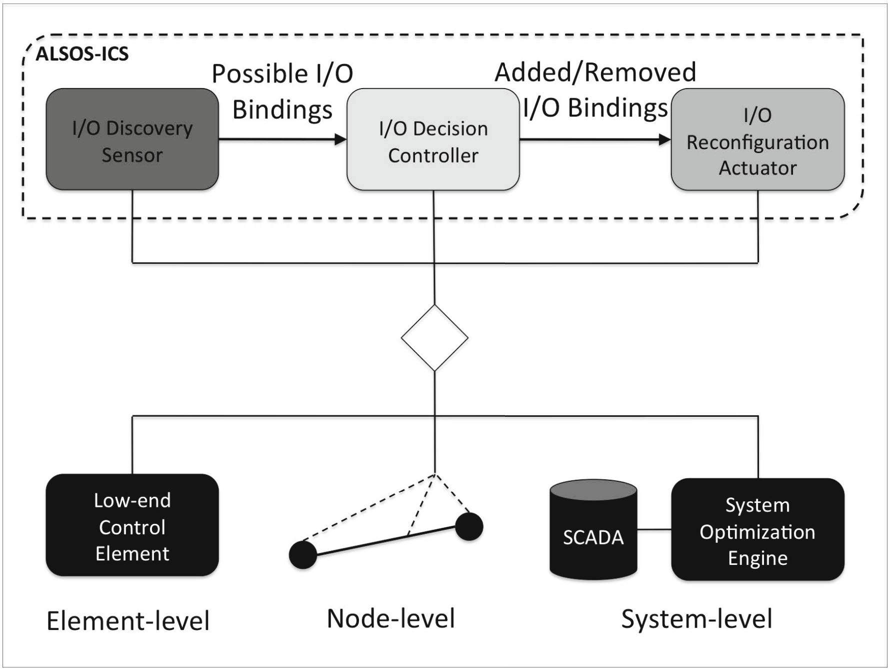


Fig. 1 Positioning of ALSOS-ICS in the context of the Smart Power Grid

ALSOS-ICS is composed of three types of control elements: (i) the *I/O discovery sensor*, (ii) the *I/O decision controller* and (iii) the *I/O reconfiguration actuator*. These elements are bound to each other and also to elements that compose a different control application. The later application is the one that experiences dynamic I/O binding capabilities provided by an ALSOS-ICS control application. The I/O discovery sensor senses for possible I/O bindings and outputs these possible bindings to the I/O decision controller. The possibility of gossip-based discovery sensors is discussed in our earlier work [29]. The I/O decision controller selects to add or remove I/O bindings based on the objectives of the control application that ALSOS-ICS supports. These objectives may be represented as a fitness function or high-level business rules and policies. The added and removed I/O bindings are the output to the I/O reconfiguration actuator that modifies the I/O binding of the served control application.

The coupling of ALSOS-ICS with control applications can be performed at different levels as illustrated in our earlier work [29]: (i) *system-level*, (ii) *node-level* and (iii) *element-level*. This book chapter shows more specifically the relevance of these levels in the Smart Power Grid. Figure 1 illustrates an overview of ALSOS-ICS and its coupling to the Smart Power Grid.

In the system-level coupling, ALSOS-ICS is linked to a Distribution Automation System (DAS) that may perform various system optimizations such as power flow optimization [5], secure fault isolation [2] etc. Data acquisition is, to an extent, centralized. A node-level deployment of ALSOS-ICS distributes I/O binding control over the Smart Power Grid at different control points such as power lines, substations, etc. Because of a higher decentralization in node-level compared to system-level, the ALSOS-ICS control elements require in this case more complex interactions that guarantee access to remote information. Finally, coupling ALSOS-ICS at the element-level introduces dynamic I/O binding control at the very low-end control elements of the Smart Power Grid. In this case, control elements reconfigure their I/O binding autonomously and in a fully decentralized fashion.

3 Application Scenarios

The Smart Power Grid may experience various (cascading) system failures or malfunctions such as overloaded power lines, failures of power lines, physical disasters, demand-supply imbalances or black-outs. These events require a broad range of system reconfigurations and stabilizations to guarantee robustness and a continuous system availability of the Smart Power Grid. Power reconfigurations that prevent system failures or minimize their impact by isolating them, require a period of time to be applied, depending on the type of reconfiguration. Time is a critical factor for the prevention of system failures or the minimization of their impact by, for example, isolating these failures. System operators cannot always make optimum decisions especially when multiple transmission lines are affected simultaneously. Automated and dynamic I/O binding reconfigurations are required to stabilize the operation of the Smart Power Grid using computational intelligence embedded in control elements.

This book chapter illustrates four application scenarios that together suggest an incremental four-level protection approach based on dynamic I/O binding reconfigurations. The purpose of these application scenarios is not to introduce a new concrete protection mechanism but to underline the importance of dynamic I/O binding requirements for the robustness of the Smart Power Grid. The four levels of Smart Power Grid reconfigurations are the following:

1. *Dynamic load-balancing of power lines*: Power flow may exceed the capacity of certain power lines when demand increases or neighboring power lines fail. Rerouting power to alternative parallel power lines requires rapid I/O binding reconfigurations to prevent cascading failures to a certain extent.
2. *Dynamic switching of power flow*: Generation, transmission and especially distribution networks are supported by multiple backup power lines and switches that provide alternative power flow of the load served by a substation but also between different substations. System failures and maintenance can be managed by dynamic and automated I/O binding reconfigurations of power lines and switches instead of manual actions by system operators.
3. *Dynamic allocation of operating reserves*: Demand-supply imbalances due to system failures or a sudden demand peak require system scaling. Operating reserves are back-up power generation that can be made available within a varied time span depending on various technical constraints. Dynamic I/O binding

reconfigurations are required to activate/deactivate operating reserves for a given system situation.

4. *Dynamic restoration after blackout*: If a system failure occurs despite the above mentioned protection actions, the power system is islanded and the restoration of the system back to its normal operation is challenging. Islands should be integrated again, generators should be activated gradually and this activation should be coordinated with the rest of the generation available in the system.

This four-level protection approach can be realized by one or more cyber-physical control applications built by three types of control elements bound in an overlay network (application graph):

- *Load sensor*: This control element monitors load information from various physical assets of the Smart Power Grid. It is bound to protection controllers to provide them with the necessary information.
- *Protection controller*: This control element coordinates the stabilizations required to guarantee the protection and robustness of the Smart Power Grid.
- *Stabilization actuator*: This control element adjusts the operation of various underlying physical assets that contribute to the stabilization of the Smart Power Grid.

The power supply required for the overlay network to function is crucial and is designed to be independent of the state in the underlying infrastructure. This can be technically achieved by the availability of dedicated small-scale backup generators or the utilization of batteries [7]. This section shows the dynamic I/O binding requirements and services that such a control application can meet and consume respectively by using ALSOS-ICS.

3.1 Dynamic Load-Balancing of Power Lines

This is the first level of dynamic reconfigurations applied for the prevention of cascading and other failures. It concerns the load-balancing of power lines positioned in parallel within the generation and transmission system. If a heavily loaded line transmits an excess power load, this excess load can be rerouted to another underloaded power line positioned in parallel.

This load rerouting is possible in two ways: (i) Shifting the phase angle between the voltages in the nodes adjacent to a power line or (ii) altering the line impedance. The first approach is technically possible via a *phase shifting transformer* device mainly used for the load-balancing of power lines [31, 32]. The second approach is possible via a *thyristor controlled series capacitor* [13, 26]. This device is mainly used for minimizing power oscillations. Other more complex devices that combine these two functionalities with additional features are proposed in literature [1, 19] resulting in improved stability of power lines.

Figure 2 illustrates the concept of load-balancing between two power lines. When the power of a line reaches its capacity of 100 units, power balancing is performed by rerouting 20 units to a line with power flow of 70 units and capacity of 130 units.

These 20 units are the result of either the alteration in the voltage phase angle or in the impedance of the power line. Note that, from an engineering point of view, these alterations can be performed rapidly. However, the balancing compensation that can be achieved is related to the technical specification of the lines and therefore this approach has limitations. These limitations are out of the scope of this book chapter and are discussed in related work [13, 26, 31, 32].

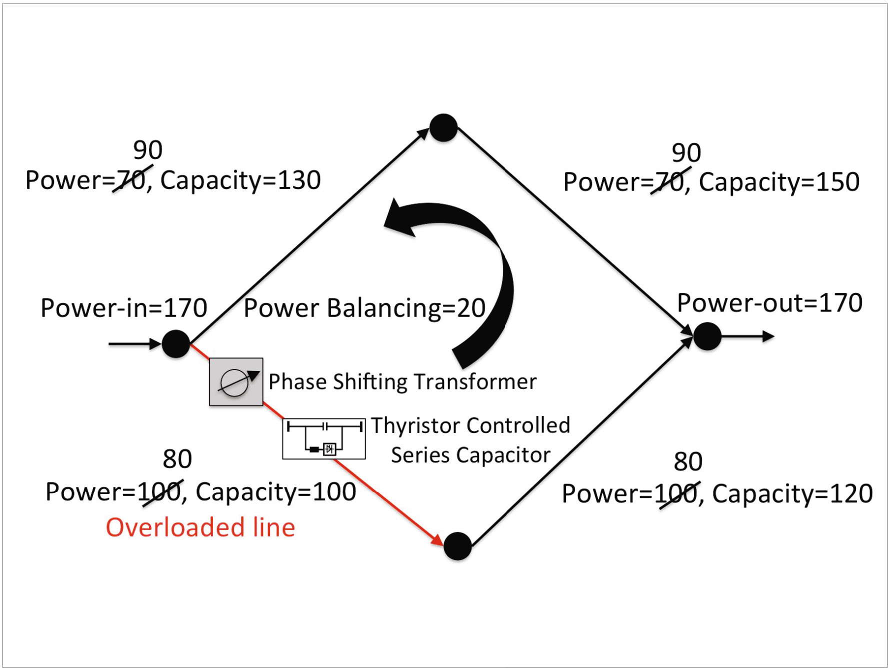

90
Power=~~70~~, Capacity=130
90
Power=~~70~~, Capacity=150
Power-in=170
Power Balancing=20
Power-out=170
Phase Shifting Transformer
Thyristor Controlled Series Capacitor
80
Power=~~100~~, Capacity=100
Overloaded line
80
Power=~~100~~, Capacity=120

Fig. 2 Load-balancing of power lines by rerouting load from an overloaded line to an underloaded one

Figure 3 illustrates the bound control elements that manage the load-balancing of transmission lines. Note that the control elements are bound in an overlay network that manages the information generated by the physical assets of the transmission lines. The load sensors in every power line output the load information to the protection controllers of the adjacent nodes. Based on this information, the balancing controllers perform decision-making about the power rerouting. This decision is executed by a stabilization actuator that controls the phase angle or the impedance of its controlled power line. More specifically, the information about the rerouted load is translated by the stabilization actuator to an actual configuration in the phase shifting transformer or the thyristor controlled series capacitor.

Note that, protection controllers require coordination. This is because a load-balancing action should not cause an overload to other power lines in the transmission topology. Therefore, the protection controllers require an on-demand dynamic

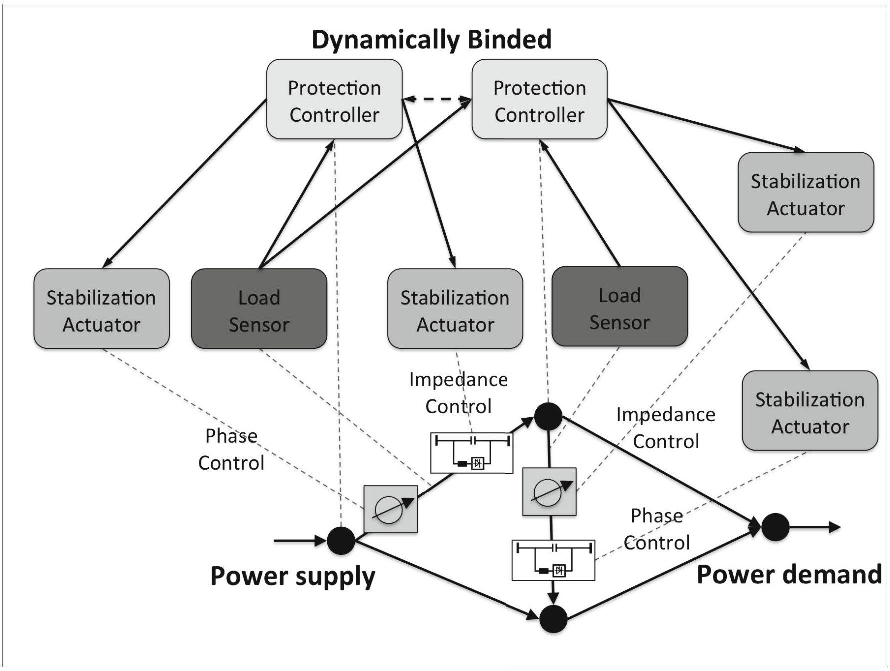


Fig. 3 The overlay network of control elements that balances the load between parallel transmission lines

I/O binding provided by an ALSOS-ICS control application. Both (i) the I/O binding control application and (ii) the load-balancing control application are realized by embedded software control elements and therefore interoperation between these two control applications is possible.

For example, assume a (tree) branch of nodes that after a load-balancing action is fed with a higher amount of power. Coordination can be performed by discovering generator nodes within this branch that can decrease their energy supply after load-balancing is performed to compensate. The protection controller of this generator node and the protection controller of the power line, in which a higher load is rerouted can dynamically bound by an ALSOS-ICS control application to negotiate and coordinate these operations. This coordination can be performed on-the-fly and in an automated fashion without the intervention of systems operators.

3.2 Dynamic Switching of Power Flow

Within (i) a generation and transmission system as well as (ii) a distribution system, power is possible to flow in different ways. This is crucial for (i) the protection of the power grid [11] and (ii) its optimization based on market strategies or economic and environmental policies [17, 25]. One way to control the power flow is by switching the power flow on or off between different nodes in a topology, e.g. in the topology of the distribution system. Switching is technically possible by using relays, switches, circuit breakers or reclosers [11]. Switching provides some form of redundancy or flexibility as power can be made available via alternative distribution paths. This provides the option to perform several critical operations [11, 27] such as (i) system clearance, maintenance, repair or construction, (ii) load-balancing and (iii) system restoration after blackouts as discussed in Section 3.3.

Figure 4 illustrates a dynamic switching scenario in a distribution system. The topology consists of nodes that represent loads, such as households, that draw power from specific feeders that are connected to substations. The control of the distribution system is hierarchical and is managed by nested control areas [11]: (i) A substation defines its control area, (ii) within which the feeders have their own control areas, (iii) within which feeder lines may also form their own control area. Therefore, the distribution system can be controlled at different granularity levels. For illustration purposes, Figure 4 focuses only on the control areas defined by the feeders.

Assume that a number of simple closed switches can transfer power between loads (i) within a control area and (ii) between different control areas. The switches are configured at 'on' or 'off' according to a system optimization [11, 17, 25]. This optimization guarantees that all loads are served and that there are no overloaded feeder lines. However, as it is mentioned before, failures may happen due to demand peaks, physical disasters or even cyber-attacks in the power grid. Failures result in power outage in one or more households and may even cascade and cause new failures in the distribution system. A coordinated switching of power flow is required to stabilize the distribution system. Note that an automated control of switches is possible via pole-top remote terminal units (RTUs) [30].

Control elements undertake this coordination by dynamically binding themselves to control the switches and, therefore, manage the power flow in a distribution system. Load sensors monitor the power flowing in a power line. This information is output to the protection controller of the feeder control area. If a failure is detected by this controller based on the input load information, an alternative power flow needs to be discovered and utilized. Protection controllers are dynamically bound and communicate to guarantee that switching of power flow does not influence other parts of the system. Next, the protection controller of the affected control area is dynamically bound with stabilization actuators that control switches within the same control area and in neighboring ones. These switches are turned on for a period of time to deliver power to the affected households. This binding configuration may last as long as the failures occurs and during repair/maintenance operations. Manual actions by system operators are not required as long as the protection controllers are able to coordinate the switching of the power flow.

Similarly to the previous scenario, the dynamic switching of power flow forms a control application that is served by an ALSOS-ICS control application. The I/O discovery sensors of ALSOS-ICS locate the feeder control areas and the switches that when closed deliver power to the affected nodes. The I/O decision controllers select the I/O binding reconfigurations required given state information provided by

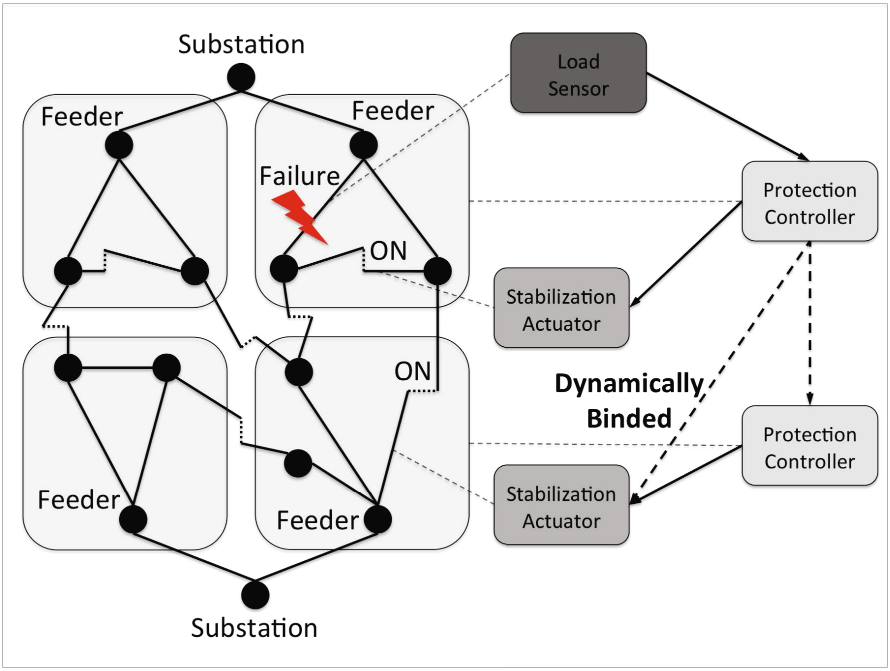


Fig. 4 Dynamic switching guarantees the power delivery in the loads of a distribution system when a failure of power lines occurs. One or more switches can close enabling a load to draw power from a neighboring control area that is served by a different feeder. Control elements are dynamically bound with each other to coordinate the dynamic switching of power flow.

the protection controllers. Finally, the I/O reconfiguration actuators of ALSOS-ICS bind the load sensors, protection controllers and stabilization actuators to handle a failure occurred in the distribution system.

3.3 Dynamic Allocation of Operating Reserves

The first two protection levels illustrated in Section 3.2 and 3.3 may not be adequate in some cases. For example, there are power lines that do not support the technology for power balancing or do not have automated switches. Furthermore, load-balancing has its limits especially when the power demand increases and additional power supply is needed in the system. Backup generation is required in the power infrastructure to match supply and demand without causing cascading failures by overloaded power lines. This backup generation is the *operating reserves* of a power grid.

Traditionally, an operating reserve is a generating power capacity available on-demand to the system operator within a period of time. An operating reserve is usually activated to meet power demand in case of system disruptions, such as power line failures or system maintenance. Although there are various operating reserves that match different system requirements, two main types of operating reserve are used by system operators [4]:

- *Spinning reserve*: This is additional synchronized generated capacity available in the system by increasing the power output of the online power generators. Spinning reserve can also refer to responsive loads as a result of demand-side energy management [21, 28].
- *Non-spinning reserve*: This is additional non-synchronized generated capacity that can be made available to the system by offline power generators within a longer period of time than spinning reserve. The power exchanged via power flow gates between different transmission zones is also a form of non-spinning reserve.

Figure 5 shows a simplified illustration of spinning and non-spinning reserves. This extra capacity can be made available within 10 minutes approximately for a period of approximately 30 minutes depending on the type of reserve and the technical features of the physical assets that enable it [6]. Failures of power lines and cascading failures can be prevented by choosing the point where the additional power is injected. For example, if the power lines adjacent in the main power supply of Figure 5 cannot support the extra power of spinning reserve, an alternative reserve that is adjacent to lines with higher capacity can be selected. Furthermore, offline power generators have a varying startup time that is also related to the actual power activated, referred to as *ramp rate* [6]. Multiple reserves can be activated and combined to ensure the robustness of power transmission. Finally note that the traditional spinning reserve is usually more expensive than the non-spinning one and therefore the cost can also be a selection factor.

Operating reserve is traditionally activated manually by system operators as agreed offline by market contracts [6]. As Smart Power Grids scale and becomes more complex and dynamic, failures and their cascading effects cannot be managed by system operators. An online, automated and dynamic allocation of operating reserves is required. Therefore, this section proposes the dynamic allocation of operating reserves by software embedded control systems. Figure 6 illustrates the control elements of a power line that is protected by three operating reserves: (i) A spinning reserve in the supply node, (ii) a second spinning reserve that acts as a responsive load enabled by a demand-side energy management mechanism and (iii) a non-spinning reserve that remains offline under normal system operation. Each power line and node, including the ones of the non-spinning reserves, have a load sensor and a protection controller respectively. Furthermore, every node that acts as an operating reserve has a stabilization actuator that activates and deactivates the operating reserve. The protection controllers, that are adjacent to an overloaded power line, check if a first-level reconfiguration is possible and adequate to balance the load of the affected power line as illustrated in Section 3.1. If the first-level reconfiguration cannot be applied, the protection controllers either activate their local reserve, if they have one, or coordinate with other remote protection controllers the allocation of their reserves.

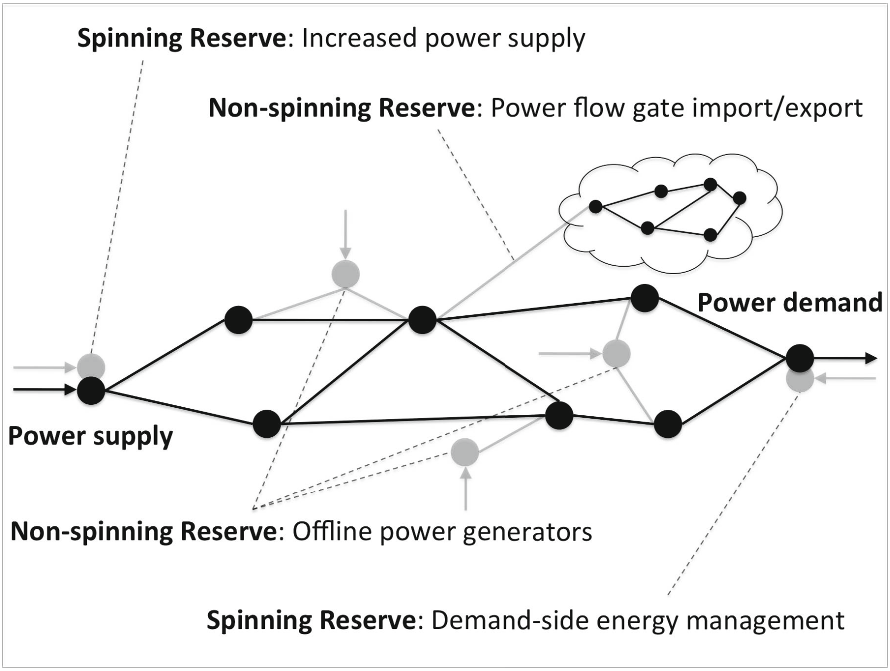

Spinning Reserve: Increased power supply
Non-spinning Reserve: Power flow gate import/export
Power demand
Power supply
Non-spinning Reserve: Offline power generators
Spinning Reserve: Demand-side energy management

Fig. 5 A power system with different operating reserves

Dynamic binding is required between the remote protection controllers to coordinate the activation of operating reserves in a cost-effective manner. Furthermore, if a non-spinning reserve is activated, the load sensor of its adjacent power lines may need to be dynamically bound with the other adjacent nodes. An ALSOS-ICS application can perform this service by inter-operating and configuring the I/Os bindings in the control application of dynamically allocated operating reserves.

3.4 Dynamic Restoration After Blackout

Despite the aforementioned levels of protection, blackouts may still occur. Blackouts in the USA have not decreased the last years and occur with higher frequency during peak times [16]. During a blackout condition, the transmission and distribution system is clustered in one or more islands that cannot exchange power due to failed power lines or nodes. These failures may be isolated or cascading. The latter is the main cause of blackouts in power systems. During a cascading failure, a power flow is forced to a rerouting that makes other lines and nodes overloaded and failing. This failing process is recursive. The system restoration after such a condition is highly complex. A simple and arbitrary restoration of the failed units does not guarantee the system restoration back to normal operation. Demand draws the power that is made available after restoration causing a new failure. Coordination is required

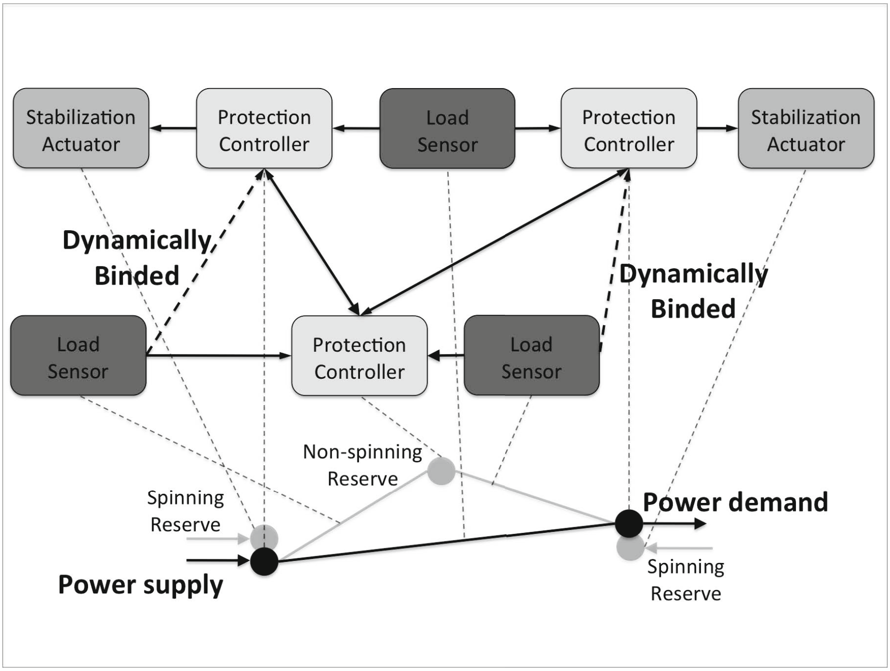


Fig. 6 The overlay network of control elements that allocate and control operating reserves for the protection of a power line from an overload

during the restoration process by interconnecting the islands and synchronizing the power flow that is made available after a blackout. This coordination means that the I/O binding of the cyber-physical control elements should be adjusted dynamically during this process. Existing restoration approaches are mainly managed by system operators that apply manual actions based on their experience [12].

Figure 7 illustrates an example of coordination performed for the restoration of a system after a blackout. Note that this scenario assumes that there is available reserved power for the control elements to perform their control tasks. Therefore, the overlay network of the control elements is energized, connected and manageable compared to the affected physical infrastructure. A simple cascading failure causes this blackout. The sequence, in which the events occur, is numbered and illustrated in order as follows: First, an unexpected failure occurs to one of the generators that causes (i) a lower power injection in the system and (ii) the unavailability of its adjacent transmission lines (event #1). Because of this failure, two events follow: (i) The power of the second generator is rerouted to its second power line and (ii) the spinning reserve of the second generator is activated. The activation occurs because of the frequency drop that the failure of the first generator causes (event #2). This increased load makes a second node overloaded and failing (event #3). This cascading failure clusters the power network in two islands. A blackout has occurred and the system needs to be restored in a coordinated and timely fashion.

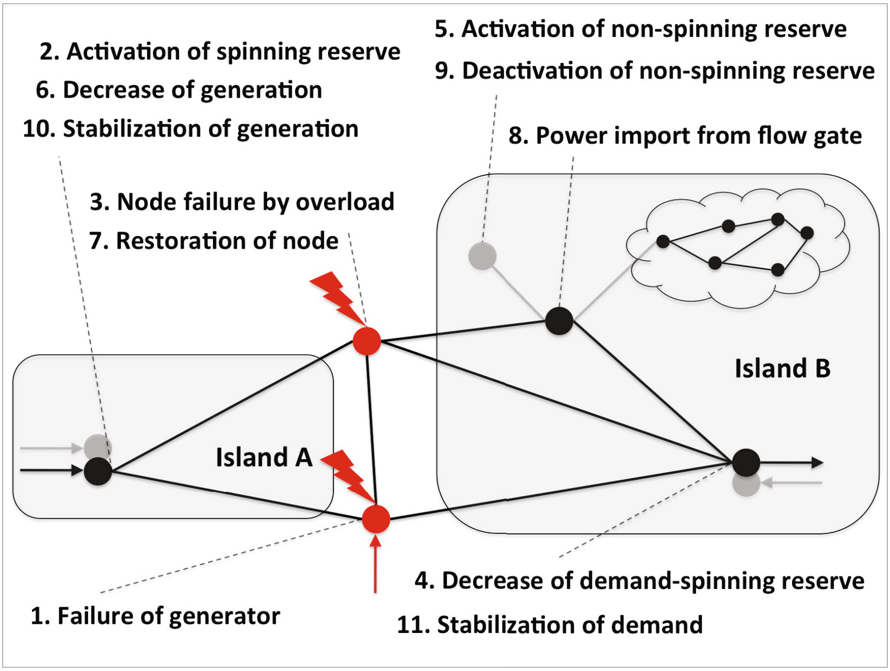

2. Activation of spinning reserve
6. Decrease of generation
10. Stabilization of generation
5. Activation of non-spinning reserve
9. Deactivation of non-spinning reserve
8. Power import from flow gate
3. Node failure by overload
7. Restoration of node
Island B
Island A
4. Decrease of demand-spinning reserve
11. Stabilization of demand
1. Failure of generator

Fig. 7 A sequence of coordination actions for a blackout restoration

The goal of the restoration strategy is to interconnect the islands that resulted after the blackout. This can be achieved by reversing the overload in the failing node. Three actions are applied: (i) Activation of spinning reserve using demand-side energy management (event #4), (ii) activation of non-spinning reserve by turning on a backup generator (event #5) and (iii) adjusting the generation in island 'A' to restore the failed node (event #6). After these actions, the islands are again interconnected (event #7). However, the activated non-spinning reserve cannot run for long and it is an expensive power source. Therefore, power is imported from a neighboring zone via a flow gate that interconnects the two zones (event #8). At the same time, the non-spinning reserve turns off (event #9), the initial power generation is stabilized (event #10) and demand is fully served again (event #11). These actions complete the system restoration. Later on, if the failed generator is fixed, new adjustments can be applied to remove the power dependency from the neighboring transmission zone.

All these actions should be executed in a certain priority and timely fashion. Synchronization is crucial. For example, the generator in island 'A' should be dynamically bound to the back-up generator in island 'B' to coordinate and adjust the allocation of power that will enable the overloaded node to be available again and interconnect the two islands. If actions are not coordinated, then new failures may occur that may result in more isolated islands that cannot be energized. For this reason, an islanding situation in the physical layer should not be reflected in the overlay network of control elements. The overlay network should remain connected and allow the control elements of ALSOS-ICS to inter-communicate between different islands and therefore support the coordination and restorations actions. Note that, as mentioned in Section 2, a gossiping protocol implementation for the I/O discovery sensors is a relevant choice in this case. Gossiping builds and maintains a dynamic well connected and non-clustered overlay network that remains robust even in case of catastrophic failures [18].

Note that, dynamic blackout restoration actions are based on dynamic load-balancing of power lines, switching of power flow and allocation of operating reserves. Therefore, more effective mechanisms for blackout prevention support a more effective blackout restoration in case it occurs.

4 Discussion

The protection of the power grid is a highly challenging and complex problem that should be decomposed and managed at different levels. This book chapter illustrates coordination scenarios within four incremental Smart Power Grid protection levels. The protection requirements in each scenario are traditionally satisfied by experienced system operators that apply manual actions assisted by centralized data acquisition and supervisory systems. However, as micro-generation scales and becomes more decentralized, such an approach becomes cost-ineffective. A higher level of automation is required, which means that control elements managing various physical assets need to become more interactive and intelligent. Distributed computational intelligence requires a situational awareness that can be attributed to a system if and only if its elements have the potential to be dynamically bound with each other on-demand. Without dynamic I/O binding of control elements, coordination cannot always be achieved. This is exactly what the four incremental scenarios for the protection of the Smart Power Grid show. For example, binding an offline generator to the rest of the control elements when operating reserves are utilized is required to control and coordinate the injected power flow in the system. Without such a binding, a protection measure may cause new cascading failures.

The modeling approach of ALSOS-ICS couples dynamic I/O binding capabilities with the rest of the control logic of a cyber-physical system. he Internet of Things requires reconfigurable physical and software control elements that ALSOS-ICS can dynamically bind and organize. ALSOS-ICS control elements are able to interoperate as a control application with the rest of the control elements. This distinction suggests a split of concerns for system developers, yet, the context remains within control systems of Internet of Things and their applications. More specifically, ALSOS-ICS developers, extending the work of application integrators, build control elements that provide dynamic I/O binding capabilities to another group of control elements, developed by domain-experts, that embed the main control application logic. In the four application scenarios illustrated in Section 3, the main domain-expert developer has knowledge about the protection of the Smart Power Grid, and more specifically about the available repair and maintenance mechanisms.

This developer provides the load-sensors, protection controllers and stabilization actuators. It assumes a certain level of interaction and communication capabilities that ALSOS-ICS developers expose via, for example, interfaces. The actual I/O binding discovery, selection and reconfiguration is handled by the control elements of ALSOS-ICS developed by network communication experts.

5 Conclusions and Future Work

This chapter illustrates application scenarios of organizational control reconfigurations for the robustness of the Smart Power Grid. Each of these scenarios requires some degree of coordination between control elements that manage various physical assets. With the emergence of micro-generation and renewable resources, matching supply and demand becomes challenging with an impact on the robustness of the power grid. Coordination needs to evolve beyond the control of system operators, become more automated, decentralized and manageable by control elements themselves. The coordination and computational intelligence of control elements requires capabilities for dynamic binding reconfigurations in this case. The application scenarios illustrated and discussed in this book chapter exactly show such required capabilities for the protection of the Smart Power Grid. Dynamic binding reconfigurations required for applications of the Internet of Things can be modeled as a control application using the ALSOS-ICS model summarized in this chapter. ALSOS-ICS allows a higher interoperation and modularity between control applications and a higher flexibility, integration and applicability of dynamic binding reconfigurations in the domains of the Internet-scale cyber-physical control systems.

Organizational control reconfigurations are required in other application domains beyond the Smart Power Grid. ALSOS-ICS is application-independent and therefore various domains of Internet of Things, such as transportation systems, air vehicles [34] etc., can make use of it.

Future work includes the actual implementation, testing and evaluation of Internet of Things applications that are based on dynamic I/O binding of their control elements. iCS [3] is a Java Micro Edition (JME) lightweight runtime environment for distributed control applications that can be used for exactly this purpose. Furthermore, a critical aspect that needs to be studied in the future Smart Power Grids is the higher interdependence of the power infrastructure to the communication infrastructure.

Acknowledgements. This research is partially funded by NLNet, the NWO project ”RobuSmart: Increasing the robustness of Smart Grids through distributed energy generation: a complex network approach” and the Pacific Northwest Smart Grid Demonstration Project.

References

1. Abdel-Moamen, M., Padhy, N.: Optimal power flow incorporating FACTS devices - bibliography and survey. 2003 IEEE PES Transmission and Distribution Conference and Exposition 2, 669–676 (2003)

2. Ahmed, M., Soo, W.: Development of customized distribution automation system (DAS) for secure fault isolation in low voltage distribution system. In: 2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1–7 (July 2008)
3. Ambrosio, R., Morrow, A., Noecker, N.: e-Business Control Systems. In: Proceedings of the 2nd International Conference on Computing, Communications, and Control Technologies, University of Texas, Austin, TX, pp. 91–96. IEEE Computer Society (2004)
4. Ch, F.D.M., Bedoya, D.B., Jannuzzi, G.D.M., Da Silva, L.C.P.: Operating reserves provided by distributed generation. In: Proceedings of the 3rd IASME/WSEAS International Conference on Energy & Environment, Stevens Point, Wisconsin, USA, pp. 219–224. World Scientific and Engineering Academy and Society, WSEAS (2008)
5. Chen, C.-S., Tsai, C.-T., Lin, C.-H., Hsieh, W.-L., Ku, T.-T.: Loading balance of distribution feeders with loop power controllers considering photovoltaic generation. IEEE Transactions on Power Systems 26(3), 1762–1768 (2011)
6. Chen, J., Thorp, J., Thomas, R., Mount, T.: Locational pricing and scheduling for an integrated energy-reserve market. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, p. 10 (January 2003)
7. Divya, K., Østergaard, J.: Battery energy storage technology for power systems an overview. Electric Power Systems Research 79(4), 511–520 (2009)
8. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid - the new and improved power grid: A survey. IEEE Communications Surveys Tutorials PP(99), 1–37 (2011)
9. Gensler, T., Zeidler, C.: Rule-Driven Component Composition for Embedded Systems. In: International Conference on Software Engineering (ICSE): Workshop on Component-Based Software Engineering (2001)
10. Georgiadis, I., Magee, J., Kramer, J.: Self-organising software architectures for distributed systems. In: Proceedings of the First Workshop on Self-healing Systems WOSS 2002, p. 33. ACM Press, New York (2002)
11. Greer, R., Allen, W., Schnegg, J., Dulmage, A.: Distribution automation systems with advanced features. In: 2011 IEEE Rural Electric Power Conference (REPC), pp. C4-1–C4-15 (April 2011)
12. Greitzer, F.L., Podmore, R., Robinson, M., Ey, P.: Naturalistic decision making for power system operators. International Journal of Human-Computer Interaction 26(2-3), 278–291 (2010)
13. Grunbaum, R., Pernot, J.: Thyristor-controlled series compensation: A state of the art approach for optimization of transmission over power links. In: 1st International Forum on Innovations in Power Links, pp. 15–20 (March 2001)
14. Guler, M., Clements, S., Kejriwal, N., Wills, L., Heck, B., Vachtsevanos, G.: Rapid Prototyping of Transition Management Code for Reconfigurable Control Systems. In: Proceedings of the 13th IEEE International Workshop on Rapid System Prototyping (RSP 2002). IEEE Computer Society, Washington (2002)
15. Hammerstrom, D., Oliver, T., Melton, R., Ambrosio, R.: Standardization of a hierarchical transactive control system. In: Proceedings of the Grid Interop 2009 Conference (2009)
16. Hines, P., Apt, J., Talukdar, S.: Large blackouts in North America: Historical trends and policy implications. Energy Policy 37(12), 5249–5259 (2009)
17. Jalilzadeh, S., Hosseini, H., Nabaei, V., Govar, G., Zandi, M.: Multipurpose reconfiguration of deregulated distribution networks using BGA. In: IEEE 2nd International Power and Energy Conference, PECon 2008, pp. 1222–1226 (December 2008)
18. Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., van Steen, M.: Gossip-based peer sampling. ACM Trans. Comput. Syst. 25(3) (August 2007)

19. Jiang, X.: Operating Modes and their Regulations of Voltage-sourced Converter Based Facts Controllers. PhD thesis, Faculty of Rensselaer Polytechnic Institute, Rensselaer Polytechnic Institute (2007)
20. Karnouskos, S.: Cyber-physical systems in the smartgrid. In: 2011 9th IEEE International Conference on Industrial Informatics (INDIN), pp. 20–23 (July 2011)
21. Kirby, B.: Load response fundamentally matches power system reliability requirements. In: IEEE Power Engineering Society General Meeting, pp. 1–6 (June 2007)
22. Koubaa, A., Andersson, B.: A Vision of Cyber-Physical Internet. In: 8th International Workshop on Real-Time Networks, RTN 2009 (2009)
23. Kramer, J., Magee, J.: Self-Managed Systems: an Architectural Challenge. In: Future of Software Engineering (FOSE 2007), pp. 259–268. IEEE (May 2007)
24. Lee, E.: Cyber physical systems: Design challenges. In: 2008 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), pp. 363–369 (May 2008)
25. Mamo, X., Mallet, S., Coste, T., Grenard, S.: Distribution automation: The cornerstone for smart grid development strategy. In: IEEE Power Energy Society General Meeting, PES 2009, pp. 1–6 (July 2009)
26. Maruf, N.I., Mohsin, A., Shoeb, A., Islam, K., Hossain, M.: Study of Thyristor Controlled Series Capacitor (TCSC) as a Useful Facts Device. International Journal of Engineering Science and Technology 2(9), 4357–4360 (2010)
27. Meier, A.: Electric power systems: a conceptual introduction. Wiley survival guides in engineering and science. IEEE Press (2006)
28. Pournaras, E., Warnier, M., Brazier, F.M.: Local Agent-based Self-stabilisation in Global Resource Utilisation. International Journal of Autonomic Computing 1(4), 350–373 (2010)
29. Pournaras, E., Yao, M., Ambrosio, R.: Dynamic composition and reconfiguration of internet-scale control systems. In: 2011 Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST), pp. 233–240 (June 2011)
30. Santos, J., Silva, N., Rodrigues, P., Rodrigues, A., Marsh, D., Gomes, F., Pinto, C.M., Blanquet, A., Carrapatoso, A.: Electric grid versus data network architectures and standards Smart Grid as plug & play. In: IET Conference Publications, 2009(CP550), p. 912 (2009)
31. Van Hertem, D., Verboomen, J., Purchala, K., Belmans, R., Kling, W.: Usefulness of DC power flow for active power flow analysis with flow controlling devices. In: The 8th IEE International Conference on AC and DC Power Transmission, ACDC 2006, pp. 58–62 (March 2006)
32. Verboomen, J., Van Hertem, D., Schavemaker, P., Kling, W., Belmans, R.: Phase shifting transformers: principles and applications. In: International Conference on Future Power Systems, p. 6 (November 2005)
33. Wang, L., Balasubramanian, S., Norrie, D.H.: Agent-based Intelligent Control System Design For Real-time Distributed Manufacturing Environments. In: Working Notes of the Agent Based Manufacturing Workshop, pp. 115–152 (1998)
34. Wills, L., Kannan, S., Sander, S., Guler, M., Heck, B., Prasad, J.V.R., Schrage, D., Vachtsevanos, G.: An open platform for reconfigurable control. IEEE Control Systems 21, 49–64 (2001)

Homesick Lévy Walk and Optimal Forwarding Criterion of Utility-Based Routing under Sequential Encounters

Akihiro Fujihara and Hiroyoshi Miwa

Abstract. The Internet of Things (IoT) is going to develop integrated and organised networks of all things and beings in the world enabling autonomous computing and information communication for the creation of new values in the future. For such networks by IoT that accept a certain level of communication delay, but that must realise highly-reliable message forwarding, Delay Tolerant Network (DTN) gives a possible solution. Recently, DTN has attracted attention as a future network under challenged network environments where communication delay, disruption, and disconnect frequently occurs. In this chapter, we review some routing protocols for efficient message forwarding in DTN. We also review some mobility models often used for simulating motions of mobile nodes to evaluate the performance of DTN. In this review, we propose our mobility model called Homesick Lévy Walk that mimics human mobility patterns of an universal scale-free property of the frequency of human contacts. After this, we also propose our utility-based routing protocol which maximises the expected number of selected relay nodes being likely to encounter a destination node under sequential encounters with nodes. We evaluate the performance of our routing protocol by comparing with some performance measures of some existing routing protocols under the condition that the Homesick Lévy Walk is adopted as mobility model. We show that our protocol is comparable to others in arrival rate of messages under a smaller number of message forwarding. We also find that the performance of our protocol is stable up to a few hundred mobile nodes and tends to be scalable with the number of nodes.

1 Introduction

The concept of the Internet of Things (IoT) aims to achieve integrated and organised networks of all things and beings in the world enabling autonomous computing and

Akihiro Fujihara · Hiroyoshi Miwa
Graduate School of Science and Technology, Kwansei Gakuin Univ., 2-1 Gakuen Sanda Hyogo 69-1337, Japan
e-mail: {afujihara,miwa}@kwansei.ac.jp

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 207–231.
DOI: 10.1007/978-3-642-34952-2_9

information communication for the creation of new values in the future. Since many things and beings are inherently mobile, IoT is inevitably needed to solve problems regarding reliable wireless communication networks between movable nodes. For such networks that accept a certain level of communication delay, disruption, and disconnect, but its communication must be highly reliable, Delay Tolerant Network (DTN) [1, 2] gives a possible solution to enhance the reliability. These networks are sometimes called Mobile Ad-hoc NETwork (MANET) or Mobile Opportunistic Network. , where dense mobile nodes form an ad-hoc network, or sparse ones communicate opportunistically when they meet each other. In recent years, DTN has attracted attention as one of possible information communication technologies capable of achieving a great performance in communication speed and reliability on the challenged environments. DTN can offer prompt and low-cost installation even in places where infrastructure is still unprepared like rural areas in developing countries, or has already been collapsed like disaster areas. If this mission of DTN is successfully accomplished, a possible practical application can be realised, i.e., a data sharing and gathering system using a vast amount of decentralized autonomous nodes which spontaneously collect environmental information. In disaster situations, for example, we think that DTN technologies enable prompt gathering of massive disaster information regarding disaster damage like dangerous areas and impassable roads that avoid prompt disaster evacuation, and also safety confirmation of victims [3, 4].

In message routing between mobile nodes in DTN, *store-carry-forward* scheme is often used to relay messages from source nodes to target nodes. An image of this scheme is illustrated in Fig. 1. Suppose that a source node A wants to forward a message (Msg.) generated in node A to a target node C. They can move, but have no chance to come close (or meet) enough to wirelessly communicate each other. As shown in Fig. 1, the store-carry-forward scheme works if they have a common node that they often meet (say, node B). By routing through a relay node B, the message can successfully forwarded. In this scheme, here, there are two important points we need to consider to achieve successful message routing: (1) mobility patterns of mobile nodes that determines which node meets which node in common, and (2) routing protocol (or message forwarding algorithm) that determines how to transfer messages efficiently and reliably.

Some mobility models have been proposed to mimic the mobility patterns of humans, animals, and vehicles for numerical simulations. Usually, mobility patterns essentially consist of uncertainty and certainty of motions. To express them, some *randomness* and *regularity* are used in mobility models. The randomness generally comes from distance, direction, and timing of moving nodes. The regularity may come from some special attraction points for nodes, such as home, workplace, and so on. Selecting an appropriate mobility model is important to reasonably evaluate the performance of routing protocols in DTN in practical situations.

On the other hand, many routing protocols in DTN have also been proposed. The most primitive one is epidemic routing [5] where each node epidemically copies all possessing messages with all encountered nodes as much as possible. This protocol makes use of all possible relay opportunities to achieve the best performance.

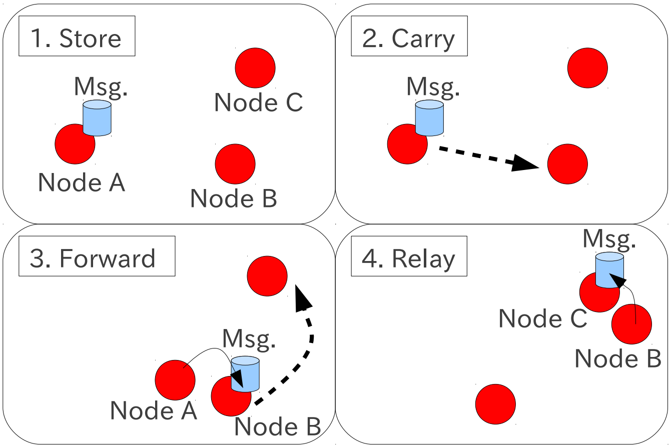


Fig. 1 An image of the store-carry-forward scheme of message routing in DTN. A message (Msg.) generated in a source node (A) can be transferred to a target node (C) via a relay node (B) using their mobility patterns.

But, the number of copied messages exponentially grows as they copies messages and the scalability of the whole network fails. In order to balance between the performance and the scalability, controls of message replication and forwarding is important. To overcome the difficulty of message flooding, there exists a class of utility-based routing protocols [6] where an utility value representing how much reachable to the destination is properly assigned to all nodes and messages are replicated in some selected nodes whose utility is relatively high.

In our previous studies [7, 8], we proposed our mobility model named *Homesick Lévy Walk (HLW)* model that simply explains human mobility patterns of an universal scale-free property of the frequency of human serendipitous encounters. The HLW model has two essential properties: long-distance travelling and homesickness. For each time step, the walker determines long-distance travelling away from home or going back to a hub of activity (home) with a certain fixed probability. After this, we also proposed our utility-based routing protocol in DTN named *Optimal Forwarding Criterion Of Utility-based Routing with Sequential Encounters (OFCOURSE)* which maximises the expected number of selected relay nodes being likely to encounter a destination node under sequential encounters with nodes. In OFCOURSE routing protocol, we used the expected value of inter-contact times between mobile nodes as the utility value, and the protocol decides which nodes to forward a message copy based on a table of forwarding criterion given by optimal stopping theory . We numerically evaluated the performance of our routing protocol by comparing four performance measures (message arrival rate, the number of forwarded messages, average hop counts of delivered messages, and average delay of delivered messages) with some existing routing protocols under the condition that the Homesick Lévy Walk is adopted as mobility model. We showed that our protocol is comparable to others in arrival rate of messages under a smaller number of message forwarding. The main contribution of this chapter is about the scalability of the OFCOURSE routing protocol. We show that the performance is stable up to a few hundred mobile nodes, and it tends to be scalable with the number of nodes.

The rest of the chapter is organized as follows. In Section 2, we review some typical mobility models often used in the studies on routing protocols in DTN. We also provide the detailed explanation of the Homesick Lévy Walk mobility model. In Section 3, we also review some well-known routing protocols in DTN. Section 4 describes the algorithm of our routing protocol in details. In Section 5, we show some results of the performance evaluation by simulating the performance measures by changing the number of mobile nodes. Finally, we summarize our work and discuss future directions of our results in Section 6 and 7, respectively.

2 Existing Mobility Models and Homesick Lévy Walk

Here, we briefly introduce typical and well-used mobility models for the context of the performance evaluation in DTN or MANET [16]. We explain them by focusing on two important properties: *randomness* and *regularity* . After this, we explain the Homesick Lévy Walk model in details.

2.1 Random Walk

Random Walk (RW) is a most simple walk model that the walker randomly determines which distance and direction to hop by certain probability distribution functions. Usually, the distance is given by an independently and identically distributed (IID) random variable whose variance is finite, and the direction is by the uniform distribution on $[0,\pi)$. The motion of this walk is uncorrelated, i.e., completely governed by randomness, but no regularity in the traces. Random walk was sometimes used for mobility patterns of humans and animals. But, some recent studies have shown that they tend to often travel longer distances which seem to be given by a fat-tailed distribution function.

In some close-range wireless communication experiments, such as Bluetooth, Wi-Fi, and ZigBee, It has been reported that inter-contact time between human carried mobile devices in a short time period (several days) obeys a truncated power-law distribution [14]. This statistical property on inter-contact times can be explained even by random walk [15]. But, this power-law distribution usually becomes unstable in longer time periods.

2.2 Random Waypoint

Random Waypoint (RWP) is a mobility model that the walker randomly determines the next destination from one's territory where one can move by a certain probability distribution (where the average distance to the next destination diverges in general) in order to go there straight (or by a shortest path) with a certain velocity.

Usually, the next destination is determined by a uniform distribution or properly weighted one covered on the territory. As a modification, the velocity can also be given by a certain probability distribution function. The motion of this walk is correlated in the sense that the walker go straight to the next destination. But, the sequence of destination positions is uncorrelated, which is the randomness of this walk.

The random waypoint model is often used for mobility patterns of vehicles. The power-law distribution of inter-contact time can also be explained by this model [15].

2.3 *Lévy Flight and Lévy Walk*

Lévy Flight (LF) is a mobility model that the walker flies to the next destination given by a power-law distribution function whose variance diverges in general. The probability distribution function is described as

$$p(l) \sim 1/l^{2+\mu}, \tag{1}$$

where $l(>0)$ is a flight length and $0 < \mu \leq 2$ is a scaling parameter. The word "Flight" means that the walker jumps directly to the next destination. When the walker gradually move to the next destination with a certain velocity, the model should be called Lévy "Walk" (LW) model.

When $\mu \leq 0$, since the average of $p(l)$ diverges, the distribution function $p(l)$ becomes the same as a uniform distribution, meaning that the motion of Lévy Walk is equivalent to that of Random Waypoint. When $\mu > 2$, since the variance of $p(l)$ becomes finite, the motion of Lévy Walk becomes that of Random Walk.

These Lévy Flight and Lévy Walk behaviour is often observed in human and animal mobility traces obtained by experiments with GSM and GPS [9, 10]. This fact indicates that moderately long-distance travelling, where the average distance is finite, but the variance of distance is divergent, is essential to the real human and animal mobility patterns.

2.4 *Homesick Lévy Walk*

Homesick Lévy Walk (HLW) is a simple extended model of Lévy Walk that we proposed to explain the frequency of serendipitous human encounters in daily life. Here, first, let us introduce the background of our proposal of this model.

Our research group has conducted a wireless communication experiment using human-carried mobile devices with Bluetooth and Wi-Fi to investigate the statistical property of contact frequency between humans in long-term periods. Today, many devices, such as smartphones, mobile PCs, portable game devices, and so on, have included Bluetooth and Wi-Fi as standard equipment and they are often moving with their possessors. By this experiment, we can obtain sampled data of the frequency of human contacts since we can assume that one detection of a device means one encounter or passing with another person. In total, eleven people participated in the

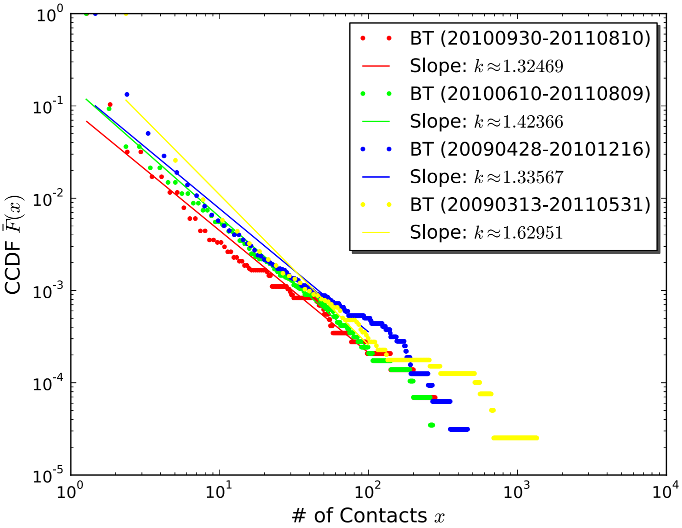


Fig. 2 Some typical frequency distributions of human contacts by Bluetooth data of selected four participants

experiment and go around as usual in daily life with carrying a PDA that scans and detects nearby Bluetooth and Wi-Fi devices to collect data every 20 seconds on average. The periods of the experiment varies with people from several months at minimum to several years at maximum in this case. The collected data consist of time stamp and MAC address of detected devices. We have analysed the statistical property of contact frequency using the collected data.

Our analysis has shown that the Complementary Cumulative Distribution Function (CCDF) of human contact frequencies universally obeys a power-law distribution [7]. Typical CCDFs are illustrated in Fig. 2. The trend of straight lines on the double logarithmic plot is a clear sign of the power-law distribution.

$$F(X \geq x) \equiv \bar{F}(x) \sim x^{-k}, \tag{2}$$

where X is a random variable of human contacts, x is the number of human contacts for a person, and k is a scaling exponent. The values of the scaling exponent seems to varies with people between one and two.

As previously mentioned, the statistical property of human mobility traces has been explained by the Lévy walk model. However, we have found that this model cannot explain the power-law of contact frequency [7]. The reason for the discrepancy mainly comes from a lack of the hub of activity, *i.e.*, home. This is essential to express the regularity in human mobility patterns, especially in the long-term human mobility traces. Therefore, we cannot neglect the effect of periodic return to home. We called this regularity *homesick property*. In this context, we have naturally proposed the Homesick Lévy Walk (HLW) model.

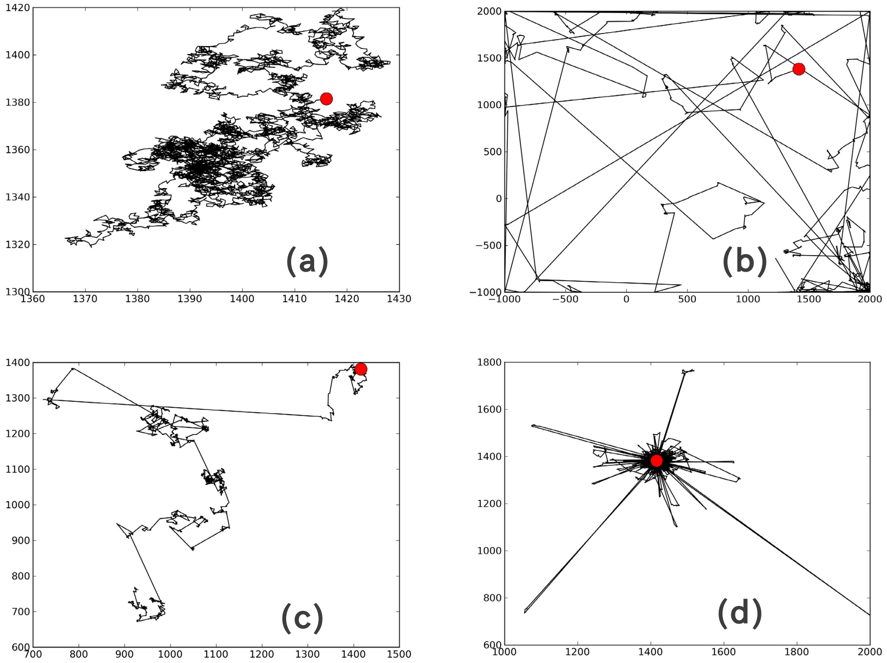


Fig. 3 A typical trajectory of (a) Random Walk, (b) Random Waypoint, (c) Lévy Walk (LW), and (d) Homesick Lévy Walk (HLW). Small (red) circles indicate the initial start point. In HLW, the circle is defined as "home," *i.e.*, the hub of activity.

Homesick Lévy Walk has two essential properties. The first one is long-distance travelling . This is defined using Lévy Walk whose flight radius r is governed by a power-law distribution with exponent $0 < \delta \leq 2$, *i.e.*,

$$p(r) \sim 1/r^{d+\delta} \tag{3}$$

where d is the spatial dimension d. In this chapter, we normally consider $d = 2$. The second property is homesickness . We defined homesickness as the decision to return home after the walker reached its destination. In HLW, the walker decides to return home with a given probability σ, otherwise it continues travelling to the next destination. For simplicity, we define the position of home as the initial position of the walker.

Note that when $\sigma = 0$, HLW becomes LW because of lack of the homesick property. Furthermore, HLW becomes RW when $\delta > 2$ and $\sigma = 0$, and also HLW becomes RWP when $\delta \leq 0$ and $\sigma = 0$. The difference between their trajectories is illustrated in Fig. 3.

We performed numerical simulations to see the contact frequency between HLWs in $d = 2$. This numerical result is illustrated in Fig. 4. As you can see, the contact frequency of HLW shows the power-law distribution in Eq. (2), while that of LW ($\sigma = 0$) decays exponentially. Therefore, we have found that HLW and LW give

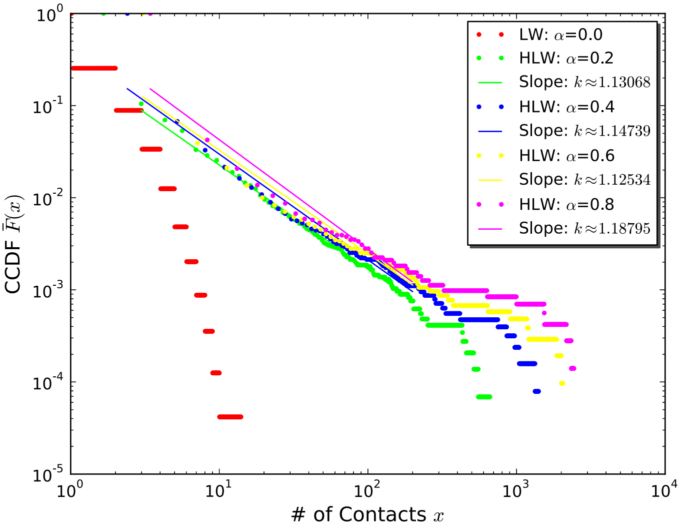


Fig. 4 Typical CCDFs for the contact frequency values $\bar{F}(x)$ employing LW($\sigma = 0$) and HLW($\sigma > 0$) in numerical simulations where the number of HLWs is $N = 10^3$, the duration of simulation time is $T = 10^5 [s]$, the scaling exponent is $\delta = 0.2$, and the homesick probability $\sigma = 0.0, 0.2, 0.4, 0.6, 0.8$ (dots) with their least squares fittings (solid line)

completely different statistical property of the contact frequency. Therefore, we can conclude that the existence of home is important to express human mobility patterns more accurately.

The Homesick Lévy Walk model can also extended to multiple return sites, for example, home and workplace. It is also interesting to consider this type of extensions, but one return site is enough to simply explain qualitative behaviour of the frequency of human contacts.

3 Routing Protocols in Delay Tolerant Networks

Here, we briefly introduce routing protocols in DTN. Since DTN usually allows communication delay, it is hard for all nodes to accurately understand rapid changes occurring among the whole network. Therefore, DTN routing protocols is often given by some heuristic algorithm in general. Many routing protocols have been proposed (See [6]), but we explain here four typical protocols: Epidemic routing, Spray and Wait, PRoPHET, and MaxProp.

By paying attention to how the protocols replicate message copies and forward them to the destination, we can categorise these protocols into two types: epidemic type and utility-based type. In epidemic type, there is no clear criterion to select which nodes to transfer messages. In general, epidemic type routing create message copies as many as possible and nodes relay them between nodes based on *first come, first forward* policy and wait for the chance that one of them happens to reach the destination. In utility-based type, on the other hand, nodes calculate some utility value to distinguish which nodes are likely to often meet, then forward message copies preferentially to nodes which have high utility values. In general, the epidemic type is effective if network resources (for example, the buffer sizes of nodes for memorising forwarded messages) are fully available. On the other hand, the utility-based type is effective if network resources are lacking or tightly limited and the loss of forwarded messages by buffer overflow should be avoided.

3.1 Epidemic Routing

Epidemic routing [5] is a routing that a node uses every communication opportunity to distribute possessing message copies to other encountered nodes if they don't possess the copies. Since message copies spread epidemically, therefore, this routing achieves the fastest message transfer to the destination. However, epidemic spreading of message copies exponentially increases the number of messages in the network, which causes network congestion if network resources are lacking or limited.

3.2 Spray and Wait

Spray and Wait [11] routing is an epidemic type routing modified by limiting the number of allowable message copies c in order to lower network resource utilisation and avoid the network congestion. There are two phases in the Spray and Wait routing: the spray phase and the wait phase. In the first phase, a node spray message copies based on the "first come, first forward" policy until the number of copies reaches the allowable number c. In the second phase, the node waits for one of sprayed messages happening to directly reach the destination. As can be seen, Spray and Wait is originally a two-hop routing. But, it is also extended to a multi-hop version where the number of message copies is saved for forwarded nodes and the sum of message copies is controlled to become equal to the maximum allowable number of message copies c in total. An image of the difference between two-hop and multi-hop versions is illustrated in Fig. 5.

3.3 PRoPHET

The Probabilistic Routing Protocol using History of Encounters and Transitivity, abbreviated as PRoPHET [12], is a utility-based protocol which utilise a history of the number of encounters with nodes as the utility function. This utility function called *delivery predictability* $P_t(i,j)$, where t is an index of time step and i,j are node indices, gives likelihood of node i meeting with node j.

Each node i calculates $P_t(i,j)$ for all $j \neq i$ when it meets node j using following equations, *i.e.*,

$$P_{t+1}(i,j) = P_t(i,j) + \alpha(1 - P_t(i,j)), \tag{4}$$

$$P_{t+1}(i,k) = P_t(i,k) + \beta(1 - P_t(i,k))P_{t+1}(i,j)P_t(j,k), \tag{5}$$

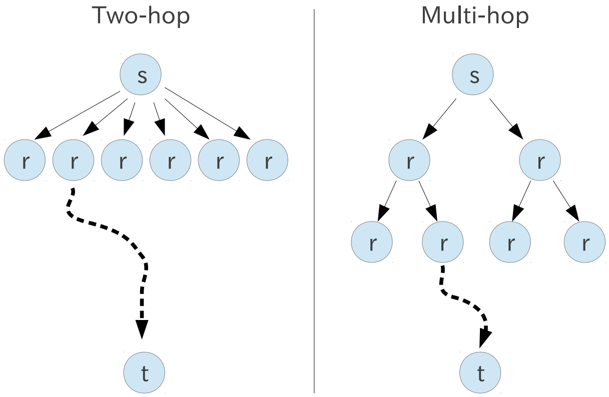


Fig. 5 Difference of forwarding message copies between two-hop and multi-hop versions when the number of allowable message copies is $c = 6$. Node s is the source node which has the original message. Node t is the target node to transfer one of message copies. Nodes r are relay nodes that message copies are forwarded from Node s. Solid and dotted allows indicate the message forwarding.

for all nodes labeled by $k \neq i, j$, where $0 \leq \alpha, \beta \leq 1$ are constants. Furthermore, for each time step, the delivery predictability gradually decreases as time passes, *i.e.*, the aging effect of $P_t(i, j)$ is described as

$$P_{t+1}(i, j) = \gamma P_t(i, j), \tag{6}$$

for all nodes labeled by $j \neq i$, where $0 \leq \gamma < 1$ is an aging constant. This equation plays a role of forgetting contact information in the distant past. In PRoPHET, the parameters are initially given by $\alpha = 0.98, \beta = 0.25, \gamma = 0.75$ in usual cases, followed by the setting of the original paper [12].

In the PRoPHET routing protocol, node i forwards message copies to encountered node j if the delivery predictability of node i to destination node l, $P(i, l)$, is higher than that of node j, $P(j, l)$. This heuristic selection mechanism based on the delivery predictability can reduce the number of message copies and contribute the network congestion.

Note that the sum of delivery predictability $P(i, j)$ over other nodes $j \neq i$ isn't always equal to one because of the aging effect. Therefore, the delivery predictability is no longer a probability value.

3.4 MaxProp

MaxProp [13] is also a utility-based protocol that estimates delivery likelihood as the utility value to find efficient relay routes with high arrival rate and low communication delay. When node i and node j meet each other, they update its contact frequency $f(i, j)$. Furthermore, a graph with n nodes is generated, where n is the number of nodes in the network, and edges are linked when $f(i, j) > 0$ and the weight of the edge between node i and j is given by $1 - f(i, j)$. Then, the delivery likelihood is calculated by the sum of weights on a shortest path from i to the destination node. Nodes select a relay path with the lowest value.

In general, the calculation of the delivery likelihood in MaxProp is harder than that of the delivery predictability in PRoPHET because MaxProp also calculates a shortest path. In some cases, however, MaxProp can keep higher performance of message delivery rate.

4 Optimal Forwarding Criterion for Utility-Based Routing under Sequential Encounters

In this section, first, we briefly introduce the optimal stopping theory that generally provides an optimal stopping time where a certain profit can be maximised. After this, we explain our routing protocol based on the optimal stopping theory. We also show the performance evaluation of our routing protocol by comparing some performance measures with those of existing routing protocols in DTN which explained in the previous section. Finally, we also mention about the scalability of our protocol by showing some numerical results with changing the number of nodes in the network.

4.1 Optimal Stopping Theory

Optimal stopping theory provides us when to make a decision for maximising the expected value of a certain profit in a time series of events governed by random variables with their profit functions. For example, the optimal stopping theory is applied for decision problems on timing of machine replacement, asset managements, employments, and financing.

The optimal stopping theory generally assumes the following two conditions.

- a time series of $n(< \infty)$ events governed by random variables: $X_1, X_2, \cdots, X_n$,
- a series of profit functions by the n events: $y_0, y_1(x_1), y_2(x_1, x_2), \cdots, y_n(x_1, \cdots, x_n) \in \mathbf{R}$.

An agent observes sequentially the series of random variables, $X_1, X_2, \cdots$. For each i step after i events $X_1 = x_1, X_2 = x_2, \cdots, X_i = x_i$ are observed, the agent makes a decision whether to stop with gaining the profit $y_i(x_1, \cdots, x_i)$ or not. Since n is finite, the agent must decide until the final n-th event.

In this case, we can solve this problem by backward induction. Since the agent must decide to stop at the final n-th event, the expected maximum profit at the n-the event V_n that we need to consider in this problem can be described by $V_n(x_1, x_2, \cdots, x_n) = y_n(x_1, x_2, \cdots, x_n)$. At the $(n-1)$-th event, the agent compares the profit gained if the stopping will be decided, $y_{n-1}(x_1, \cdots, x_{n-1})$, with the expected value of profit that will be gained if the stopping isn't decided, $E(V_n(x_1, \cdots, x_{n-1}, X_n) | X_1 = x_1, \cdots, X_{n-1} = x_{n-1})$. Therefore, the expected maximum profit V_{n-1} at the $(n-1)$-th event is described as follows.

$$V_{n-1}(x_1,\cdots,x_n) = \max\{y_{n-1}(x_1,\cdots,x_{n-1}), \\ E(V_n(x_1,\cdots,x_{n-1},X_n)|X_1 = x_1,\cdots,X_{n-1} = x_{n-1})\}. \quad (7)$$

For the (i)-th event ($1 \le i \le n-2$), the expected maximum profit $V_i(x_1,\cdots,x_i)$ satisfies the following relation as well.

$$V_i(x_1,\cdots,x_i) = \max\{y_i(x_1,\cdots,x_i), \\ E(V_{i+1}(x_1,\cdots,x_i,X_{i+1})|X_1 = x_1,\cdots,X_i = x_i)\}. \quad (8)$$

The equation (8) is called the optimality equation. As can be seen, we can calculate all V_i for $1 \le i \le n$ using Eq. (8) recursively, like the dynamic programming. Here, we can define the optimal stopping time as the first time i_{opt} when the expected maximum profit becomes $V_{i_{opt}}(x_1,\cdots,x_{i_{opt}}) = y_{i_{opt}}(x_1,\cdots,x_{i_{opt}})$, or equivalently $y_{i_{opt}}(x_1,\cdots,x_{i_{opt}}) \ge E(V_n(x_1,\cdots,x_{n-1},X_n)|X_1 = x_1,\cdots,X_{n-1} = x_{n-1})$. Therefore,

$$i_{opt} = \min\{(1 \le)i(\le n) : V_i(x_1,\cdots,x_i) = y_i(x_1,\cdots,x_i)\}, \quad (9)$$

is the optimal stopping time.

Ano and Tamaki [17] generalised this problem to that of maximising the expected maximum profit (or probability) when in total $m(\ge 2)$ stopping decisions are allowed in a time series of n events. First, they considered the following three values.

- $V_i^{(m)}$: the expected maximum profit at the i-th event under the condition that more m stopping decisions are allowed.
- $U_i^{(m)}$: the expected probability at the i-th event if the agent make a stopping decision and will obtain the best profit of all under the condition that more m stopping decisions are allowed.
- $W_i^{(m)}$: the expected probability at the i-th event if the agent doesn't make a stopping decision and will obtain the best profit of all under the condition that more m stopping decisions are allowed.

In order to solve this problem, they also derive the optimality equation as follows.

$$U_i^{(m)} = \frac{i}{n} + W_i^{(m-1)}, \quad (10)$$

$$W_i^{(m)} = \sum_{j=i+1}^{n} \frac{i}{(j-1)} V_j^{(m)}, \quad (11)$$

$$V_i^{(m)} = \max\left\{U_i^{(m)}, W_i^{(m)}\right\}, \quad (12)$$

for $i = 1,2,\cdots,n-1$. Using the following initial values $W_i^{(0)} = V_i^{(0)} = 0$ for $i = 1,\cdots,n$ and $V_n^{(m)} = 1$ for $m = 1,\cdots,n$, we can calculate the three values for all n and m In this case, the optimal stopping time under the condition that more m stopping decisions are allowed is defined by $i = i_{opt}^{(m)}$ with first satisfying $U_i^{(m)} \ge W_i^{(m)}$, or equivalently,

$$i_{opt}^{(m)} = \min\{i : U_i^{(m)} \geq W_i^{(m)}\}. \tag{13}$$

We apply this strategy of the optimal stopping theory to our routing protocols in DTN by replacing the profit with the probability of encountering a node for making a decision whether to forward messages.

4.2 Proposed Routing Method

We consider a network with $n+1$ nodes. We assume that each node experiences n encounters with the other nodes in a time series and makes c decisions to forward message copies based on the optimal stopping theory . Our proposed method is a utility-based routing whose utility is given by the expected value of inter-contact times with the other nodes. The core idea of our routing protocol is to utilise the results by Ano and Tamaki [17] in order to maximise the expected number of c selected nodes being likely to reliably reach the copies to destination nodes.

Our method can be explained by the following three steps. The first step is to introduce the calculation of the expected value of inter-contact times for each node using the history of encounters. The second step is to explain how to create an optimal forwarding criterion for selecting nodes with a high-ranked utility. The final step is to show an algorithm to forward message copies based on the utility values and the optimal forwarding criterion.

Step 1. Calculation of Utility Values

Let $I_{i,j}(t)$ be the number of encounters between nodes labeled by i and j with inter-contact time t, and $\tau_{i,j}$ be the elapsed time from the last encounter with nodes i and j to the present. By using $I_{i,j}(t)$ and $\tau_{i,j}$, the expected value of inter-contact times is defined as follows.

$$E_{\tau_{i,j}}[t] = \frac{\sum_{t\geq 0} t I_{i,j}(t+\tau_{i,j})}{\sum_{t\geq 0} I_{i,j}(t+\tau_{i,j})} \tag{14}$$

Node i calculates $E_{\tau_{i,j}}[t]$ as the utility values at every encounter with node j for $1 \leq j \leq n, j \neq i$.

Step 2. Calculation of the Optimal Forwarding Criterion

We consider $P_{i,r'}^{n,r}$ as the probability that the i-th encountered node is at the r'-th place in the ranking of utility values now and will be finally at the r-th place after meeting all n nodes, where $1 \leq i \leq n$ and $1 \leq r' \leq r$. Assume that, for simplicity, the chance of encountering a node is equally probable. Then, $P_{i,r'}^{n,r}$ can be calculated by the complementary event of $P_{i,r'}^{n,r}$, *i.e.*, $\bar{P}_{i,r'}^{n,r}$.

As shown in Fig. 6, $\bar{P}_{i,r'}^{n,r}$ can be interpreted as the sum of the probabilities of independent events that the first $i-1$ encountered nodes include l nodes which are finally within the $m-$th place for all $0 \leq l \leq r'-1$. Therefore, $P_{i,r'}^{n,r}$ is described by

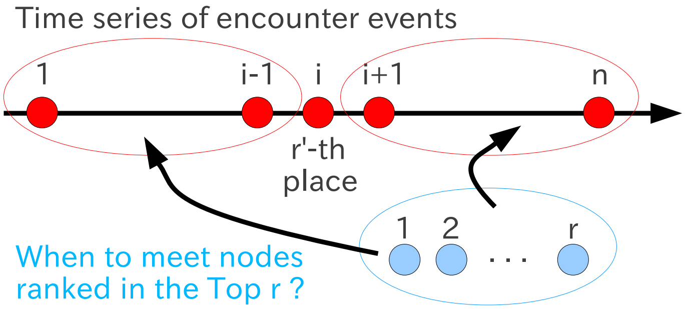

Time series of encounter events
1
i-1
i
i+1
n
r'-th place
1
2
· · ·
r
When to meet nodes ranked in the Top r ?

Fig. 6 An image for the calculation of $\bar{P}^{n,r}_{i,r'}$

$$P^{n,r}_{i,r'} = 1 - \bar{P}^{n,r}_{i,r'} \tag{15}$$
$$= 1 - \sum_{l=0}^{r'-1} {}_rC_l \frac{(i-l)_l((n-r)-(i-l-1))_{i-l}}{(n-i+1)_i},$$

for $1 \le r' \le r$, where $(a)_i = a(a+1)\cdots(a+(i-1))$.

We consider an optimisation problem of selecting at most k message forwarded nodes $i_1, \cdots, i_k$ in order to maximize the total sum of $P^{n,r}_{i,r'}$, *i.e.*,

$$\max_{1 \le i_1,\cdots,i_k \le n} \left\{ P^{n,r}_{i_1,r'} + \cdots + P^{n,r}_{i_k,r'} \right\}. \tag{16}$$

To solve this problem, we introduce a set of quantities,

$$(U^{n,r}_{i,r',k}, W^{n,r}_{i,k}, V^{n,r}_{i,r',k}),$$

for $1 \le i \le n$, $1 \le r' < r$, $0 \le k \le c$. $U^{n,r}_{i,r',k}$ is the expected value that $i-$th encountered node will be within r-th place in the whole n sequence of encountered nodes when k copies of message are still allowed and a message is forwarded to node i with the r'-th place now. $W^{n,r}_{i,k}$ is the expected value that $i-$th encountered node will be within r-th place in the whole n sequence of encountered nodes when k copies of message are still allowed and a message is not forwarded to node i. $V^{n,r}_{i,r',k}$ is defined by the maximum of the above two expected values $U^{n,r}_{i,r',k}$ and $W^{n,r}_{i,k}$.

By definition, $U^{n,r}_{i,r',k}$ and $W^{n,r}_{i,k}$ are given at $i = n$ as follows.

$$U^{n,r}_{n,r',k} = \begin{cases} 1 & (r' \le r \text{ and } r' > 0) \\ 0 & (r' > r \text{ or } r' = 0) \end{cases},$$
$$W^{n,r}_{n,k} = 0,$$

By using these as initial values, the original problem can be solved by calculating the following recursive relations.

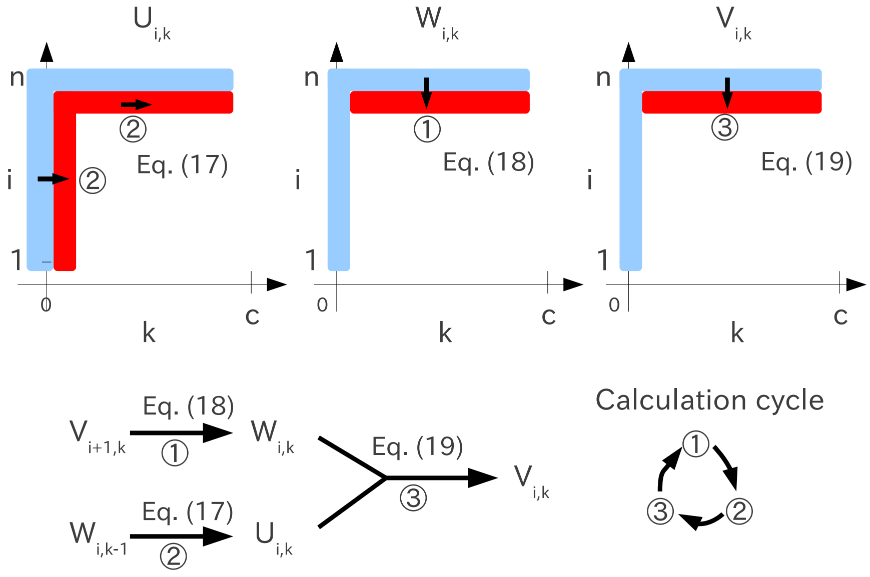


Fig. 7 The order of recursive calculations for evaluating the set $(U_{i,k}, W_{i,k}, V_{i,k})$. Here, for simplicity, we abbreviate the original set $(U^{n,r}_{i,r',k}, W^{n,r}_{i,k}, V^{n,r}_{i,r',k})$ as $(U_{i,k}, W_{i,k}, V_{i,k})$.

Table 1 Examples of the criterion

r'	$n = 10, r = 4, c = 2$	$n = 30, r = 6, c = 5$
1	Forward if $i \geq 3$	Forward if $i \geq 4$
2	Forward if $i \geq 5$	Forward if $i \geq 8$
3	Forward if $i \geq 6$	Forward if $i \geq 11$
4	Forward if $i \geq 7$	Forward if $i \geq 13$
5	Do not Forward	Forward if $i \geq 16$
6	Do not Forward	Forward if $i \geq 17$

$$U^{n,r}_{i,r',k} = P^{n,r}_{i,r'} + W^{n,r}_{i,k-1} \tag{17}$$

$$W^{n,r}_{i,k} = \frac{1}{i+1} \sum_{l=1}^{i+1} V^{n,r}_{i+1,l,k} \tag{18}$$

$$V^{n,r}_{i,r',k} = \max\{U^{n,r}_{i,r',k}, W^{n,r}_{i,k}\} \tag{19}$$

In these recursive equations, we can see the decrement of index $i+1 \rightarrow i$ in Eq. (5) and the increment of $k-1 \rightarrow k$ in Eq. (4), therefore these values can be calculated backwardly with i and forwardly with k, respectively. This recursive image is shown in Fig. 7.

Remember that $U^{n,r}_{i,r',k} \geq W^{n,r}_{i,k}$ satisfies initially at $i = n$. Also, $U^{n,r}_{i,r',k}$ decreases and $W^{n,r}_{i,k}$ increases gradually with solving backwardly with i in general. Therefore, there exists a point i^* such that $U^{n,r}_{i,r',k} \leq W^{n,r}_{i,k}$ for $i \leq i^*$. This point indicates the starting point to forward messages to node $i \geq i^*$ in the optimal strategy because $U^{n,r}_{i,r',k} \geq W^{n,r}_{i,k}$ means that forwarding to node i is better than not forwarding by definition. Two examples of the optimal forwarding criterion when $n = 10, r = 4, c = 2$ and $n = 30, r, c = 5$ are shown in Table 1.

We briefly mention the order of calculation to generate the table of the optimal forwarding criterion. To do this, we need to calculate all the values of $(U^{n,r}_{i,r',k}, W^{n,r}_{i,k}, V^{n,r}_{i,r',k})$ for $1 \leq i \leq n$, $1 \leq r' \leq r$, $0 \leq k \leq c$, and $1 \leq r \leq n$, which is $O(3cn^3)$ values in total because the calculations should be done for all the four subscripts i, r', k, r. As shown in Eqs. (4) and (6), it needs one addition or one maximization for $U^{n,r}_{i,r',k}$ or $V^{n,r}_{i,r',k}$, respectively. As shown in Eq. (5), however, it needs at most n additions and one division for $W^{n,r}_{i,k}$. Therefore, the major bottleneck for generating the table is to calculate $W^{n,r}_{i,k}$, and the order of calculation is consequently given by $O(3cn^4)$. This is a large amount of calculation when n is large, so we have to control n properly. We also mention that since the above calculations are independent of utility values, we can create the table beforehand to reduce load in each sensor node.

Step 3. Forwarding Algorithm

Here we show an algorithm to select forwarding nodes in consideration of both the expected value of inter-contact times and the optimal forwarding criterion. The inputs of our algorithm are n, r, c and $U^{n,r}_{i,r',k}, W^{n,r}_{i,k}$ for all $1 \leq i \leq n$, $1 \leq r' \leq r$, and $0 \leq k \leq c$. $U^{n,r}_{i,r',k}, W^{n,r}_{i,k}$ are calculated beforehand to use in the algorithm. We also use C as a sorted sequence of utility values in ascending order and v_i as a expected value of inter-contact times with nodes i. A pseudo code of the algorithm is described as follows.

```
Require: $n, r(\leq n), c$: Integers; $U^{n,r}_{i,r',k}, W^{n,r}_{i,k}$ : Criterion
Ensure: Optimal forwarding
  $i \leftarrow 1$
  $C \leftarrow \phi$
  while $i \leq n$ and $k > 0$ do
    if Encounter with node $i$ then
      Recalculate $v_i$
      Insert $v_i$ to $C$ and sort $C$ in ascending order
      $r' \leftarrow$ (Rank order of $v_i$)
      if $U^{n,r}_{i,r',k} \geq W^{n,r}_{i,k}$ then
        Forward message to node $i$
        $k \leftarrow k - 1$
      end if
      $i \leftarrow i + 1$
    end if
  end while
```

Note that the number of encountered nodes n can be estimated by calculating the average frequency of encounters in a certain given time T. Although we take into consideration the optimization of only single-hop routing in our algorithm, we can easily extend our method to the multi-hop routing by adding the phase of exchanging the utility sequence C with other nodes. When node i meets node j and have the utility sequence of node j, node i forwards a message to node j if the utility value of node i to the destination node is less than that of node j to the destination node, which is the simple multi-hop version of OFCOURSE. To avoid message flooding, in this case, we also take into consideration that the number of message copies is up to a given value c.

5 Performance Evaluation

In this section, we compare the performance of the multi-hop version of OFCOURSE to that of other well-known routing protocols in DTN: Epidemic routing, Spray and Wait routing , PRoPHET, and MaxProp. using some performance measures. We investigated the changes of these measures by the buffer size of nodes under the condition that the Homesick Lévy Walk model is adopted as the mobility model.

5.1 Evaluation Measures

We consider the following four performance measures for the comparison.

- Message arrival rate:
  The ratio of the number of message copies that finally arrive at destination nodes to the total number of generated messages,

- The number of forwarded messages:
  The total number of forwarded message copies by nodes,

- Average hop counts:
  The averaged hop (forwarding) counts of messages between the source node to the destination node over the ones which are successfully forwarded to the destination,

- Average delay:
  The averaged elapsed time after the messages are generated over messages which are successfully forwarded to the destination.

5.2 Simulation Setup

We consider the cases that $n = 50, 200$ nodes on the $d = 2$ dimensional space with 100×200 rectangular area. We adopt HLW as the mobility model. The nodes are randomly distributed on the area as the initial condition and the initial position of the node is assumed to be the home of HLW. The parameter δ in Eq. (3) which influences the distance to the destination is independently given by a fixed value assigned to each node and is determined by the uniform distribution on the interval [1.5,2). We consider the case that the return probability to home are $\sigma = 0.0, 0.2, 0.5, 0.8$. The velocity of all mobile nodes is randomly selected from the uniform distribution on [0,3). The communicable range of nodes where nodes can forward message copies is fixed with the radius $r_d = 1.5$.

Some additional parameter for the protocols are summarised as follows. In Spray and Wait routing [11], the maximum allowable number of message copies $c = 5$. In PRoPHET, parameters for calculating the delivery predictability are $\alpha = 0.98, \beta = 0.25, \gamma = 0.75$, which is fixed with the same values by the original paper [12]. In OFCOURSE, the maximum allowable number of message copies $c = 5$, the rank of nodes $r = \min\{(c+n)/2, n\}$, and the time steps of forwarding messages $T = 4,000$.

The Duration time of simulations is in total $15,000$ steps which consist of the following three phase. The first $10,000$ steps is for the learning phase to collect data regarding the mobility pattern of nodes only for PRoPHET, MaxProp, and OFCOURSE Routing protocols (There is no learning phase in Epidemic and Spray and Wait routing protocols). The next $4,000$ steps are for message generation and routing phase where all the nodes generate a message at each 400 steps, and the rest $1,000$ steps just for the routing phase.

In the message generation phase, the size of message created by nodes is one and the message is stored to each buffer of nodes. As the method of buffer management, we adopted First-In, First-Out (FIFO) meaning that when the buffer is full and a new message comes, it overwrites the new one on the oldest message in the buffer. Therefore, a message fails to relay to a destination node if and only if all the messages copied and distributed to buffers of nodes are overwritten by aging.

5.3 Simulation Results

Numerical results in relations between the buffer size and the arrival rate to destination nodes for each routing protocol are shown in Fig. 8. In general, the arrival rates of all the protocols increase with increasing the buffer size. We can see that the arrival rates of messages gradually degrade as increasing σ. However, the performance of OFCOURSE is relatively improved more than the others under stronger homesickness.

Numerical results in relations between the buffer size and the number of forwarded messages on ways to destination nodes in the network are shown in Fig. 9. We can see that although the number of forwarded messages in the OFCOURSE routing is almost as small as that of the Spray and Wait routing, those of PRoPHET and MaxProp are comparable in the arrival rates, especially in $n = 200$. In this sense, the performance of OFCOURSE is more efficient than that of the other utility-based routing protocols.

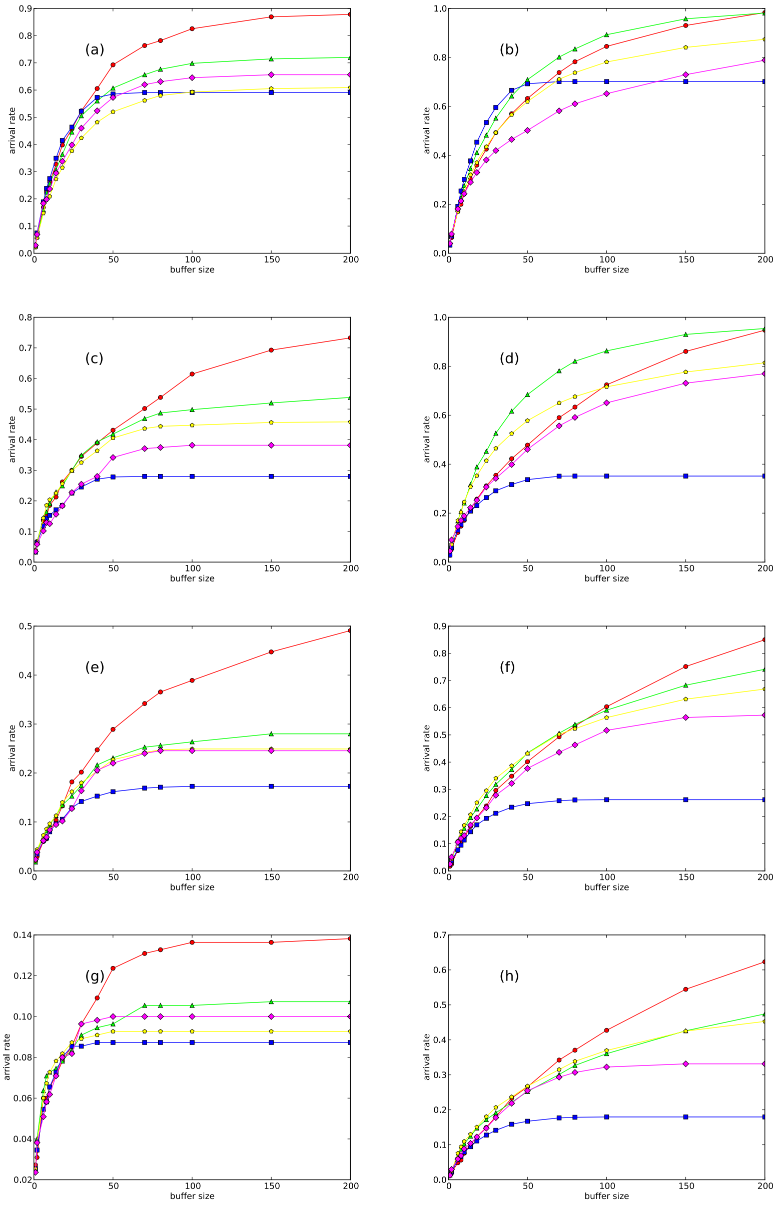


Fig. 8 Relations between the buffer size and arrival rate where the number of nodes and the return probability to home are fixed respectively as follows. (a) $n = 50$, $\sigma = 0$ (LW), (b) $n = 200$, $\sigma = 0$ (LW), (c) $n = 50$, $\sigma = 0.2$ (HLW), (d) $n = 200$, $\sigma = 0.2$ (HLW), (e) $n = 50$, $\sigma = 0.5$ (HLW), (f) $n = 200$, $\sigma = 0.5$ (HLW), (g) $n = 50$, $\sigma = 0.5$ (HLW), (h) $n = 200$, $\sigma = 0.8$ (HLW). The markers in the legend are that Epidemic (red and circle), Spray and Wait (blue and square), PRoPHET (green and triangle), MaxProp (yellow and pentagon), and OFCOURSE (pink and diamond).

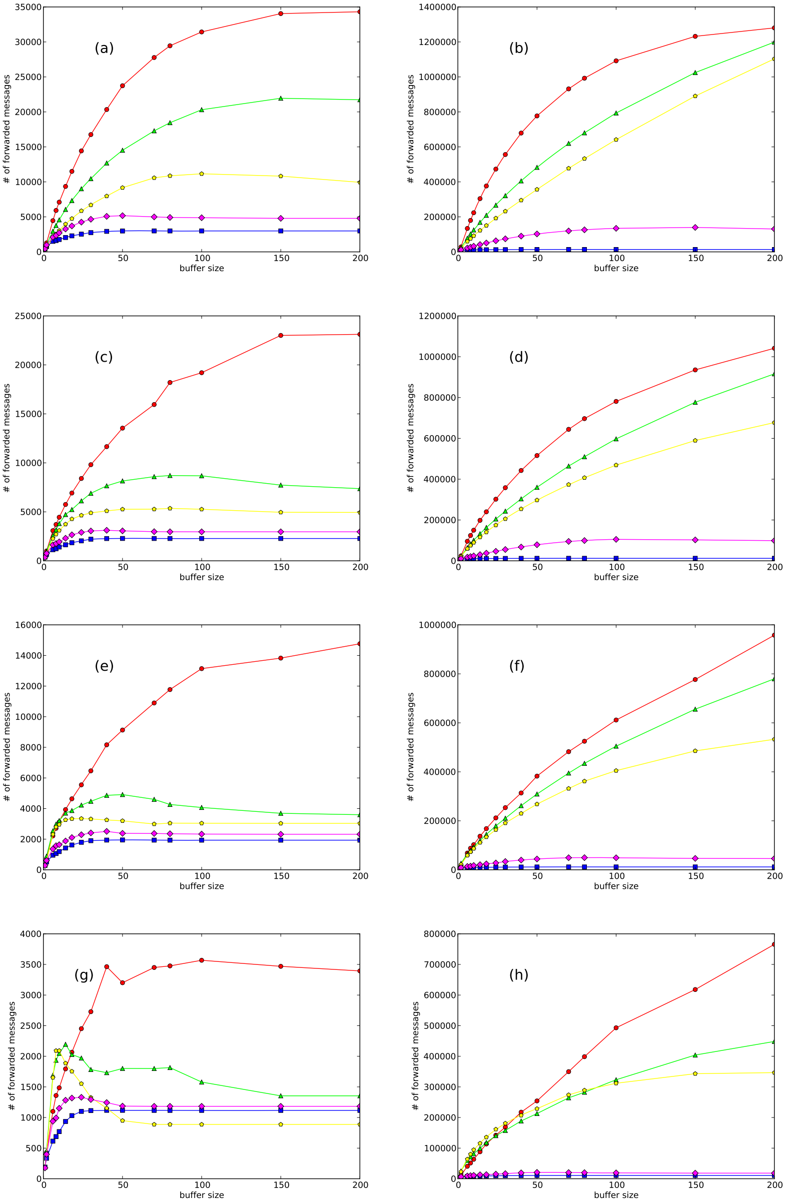


Fig. 9 Relations between the buffer size and the number of forwarded messages where the number of nodes and the return probability to home are fixed respectively as follows. (a) $n = 50$, $\sigma = 0$ (LW), (b) $n = 200$, $\sigma = 0$ (LW), (c) $n = 50$, $\sigma = 0.2$ (HLW), (d) $n = 200$, $\sigma = 0.2$ (HLW), (e) $n = 50$, $\sigma = 0.5$ (HLW), (f) $n = 200$, $\sigma = 0.5$ (HLW), (g) $n = 50$, $\sigma = 0.5$ (HLW), (h) $n = 200$, $\sigma = 0.8$ (HLW). The markers in the legend are that Epidemic (red and circle), Spray and Wait (blue and square), PRoPHET (green and triangle), MaxProp (yellow and pentagon), and OFCOURSE (pink and diamond).

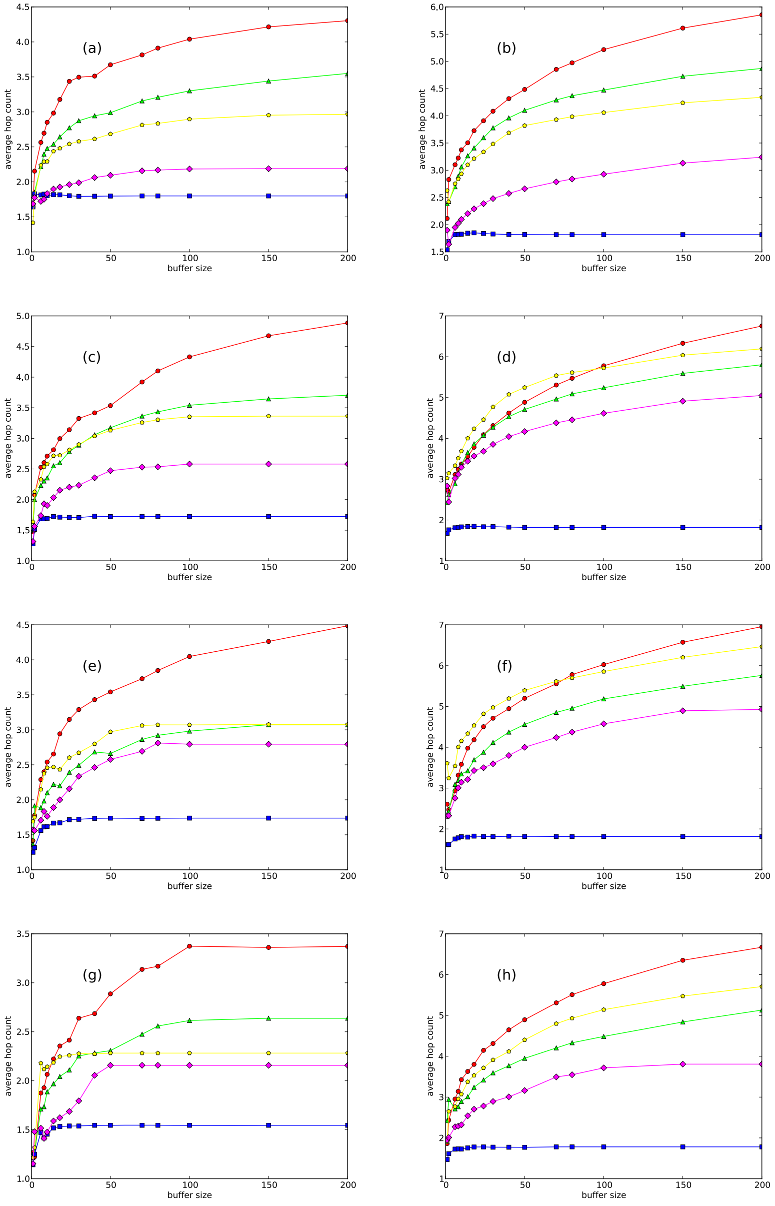


Fig. 10 Relations between the buffer size and the average hop counts where the number of nodes and the return probability to home are fixed respectively as follows. (a) $n = 50$, $\sigma = 0$ (LW), (b) $n = 200$, $\sigma = 0$ (LW), (c) $n = 50$, $\sigma = 0.2$ (HLW), (d) $n = 200$, $\sigma = 0.2$ (HLW), (e) $n = 50$, $\sigma = 0.5$ (HLW), (f) $n = 200$, $\sigma = 0.5$ (HLW), (g) $n = 50$, $\sigma = 0.5$ (HLW), (h) $n = 200$, $\sigma = 0.8$ (HLW). The markers in the legend are that Epidemic (red and circle), Spray and Wait (blue and square), PRoPHET (green and triangle), MaxProp (yellow and pentagon), and OFCOURSE (pink and diamond).

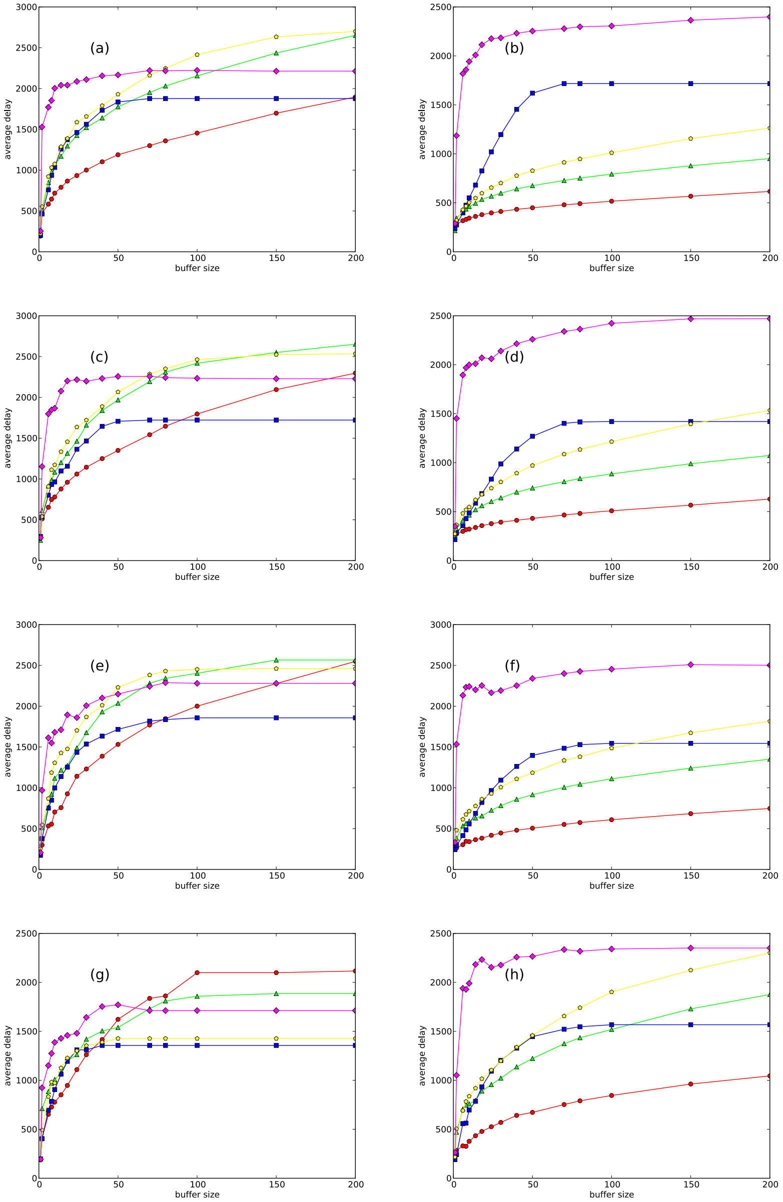


Fig. 11 Relations between the buffer size and the average delay time where the number of nodes and the return probability to home are fixed respectively as follows. (a) $n = 50$, $\sigma = 0$ (LW), (b) $n = 200$, $\sigma = 0$ (LW), (c) $n = 50$, $\sigma = 0.2$ (HLW), (d) $n = 200$, $\sigma = 0.2$ (HLW), (e) $n = 50$, $\sigma = 0.5$ (HLW), (f) $n = 200$, $\sigma = 0.5$ (HLW), (g) $n = 50$, $\sigma = 0.5$ (HLW), (h) $n = 200$, $\sigma = 0.8$ (HLW). The markers in the legend are that Epidemic (red and circle), Spray and Wait (blue and square), PRoPHET (green and triangle), MaxProp (yellow and pentagon), and OFCOURSE (pink and diamond).

Numerical results in relations between the buffer size and the average hop counts until messages arrive at the destination node are shown in Fig. 10. Our routing protocol always keeps smaller average hop counts than other routing protocols except the Spray and Wait protocol which is two-hop routing.

Numerical results in relations between the buffer size and the average delay time until messages arrive at the destination node are shown in Fig. 11. A weak point of the OFCOURSE routing is the longest delay of all. This result comes from waiting to make best decisions for which node to forward message copies based on the optimal forwarding criterion . If T becomes larger, the protocol needs more time to wait. This indicates that the appropriate parameter selection of T (or identically n) is essential to ensure the performance in the OFCOURSE routing.

6 Conclusion

In this chapter, we investigated some properties of the OFCOURSE routing protocol as an effective store-carry-forward and utility-based routing protocol in DTN by using the expected value of inter-contact times as its utility value and the optimal stopping theory. The optimal forwarding criterion which enables to select at most c nodes in order to maximize the expected number of selected nodes being within the $r-$th place and forward message copies reliably to the destination nodes. As a realistic mobility model, we adopted Homesick Lévy Walk which explains the scale-free properties of both travelling distances and the contact frequency observed in our experimental data. We compared the performance of our protocol with other well-known epidemic and utility-based type routing protocols. Although the OFCOURSE protocol has longer delay for relaying message copies to the destination nodes in general, it also achieve the comparable performance in the message arrival rate, keeping the better performances in the number of forwarded messages and the average hop counts compared to the others as increasing the number of nodes n, which means that the OFCOURSE routing protocol tends to have the good scalability as it is.

The Spray and Wait routing protocol may be competitive to the OFCOURSE protocol. However, the main difference between them is the order of the forwarding and wait phases. For the Spray and Wait protocol, forward first, then wait. For the OFCOURSE protocol, on the other hand, first wait for learning, then forward based on the optimal forwarding criterion. The performance of the Spray and Wait protocol depends on the instance of mobility patterns because it is not always true if some nodes that meet earlier become better relay nodes to destination nodes. In our protocol, due to the optimal stopping theory, the performance of the arrival rate is averagely stable for any instance. We think that this is the advantage of our protocol. This is also true in the multi-hop Spray and Wait protocol.

We hope that the OFCOURSE routing protocol will contributes to the future network systems in the concept of IoT for achieving integrated and organised information communications between all movable things and beings in the world.

7 Open Issues

It is necessary for future work to consider some better method to determine the parameters in the OFCOURSE protocol, such as n (or identically T), r, and c appropriately according to DTN environments. This is because the delay of relaying message copies might become quite long. Possibility of controlling the delays by choosing appropriate parameter values in the optimal forwarding criterion should be investigated in details.

References

1. Farrell, S., Cahill, V.: Delay- and Disruption-Tolerant Networking. Artech House (2006)
2. Vasilakos, A., Zhang, Y., Spyropoulos, T.V.: Delay Tolerant Networks: Protocols and Applications. Wireless Networks and Mobile Communications Series. CRC Press (2012)
3. Fujihara, A., Miwa, H.: Real-time Disaster Evacuation Guidance using Opportunistic Communications. In: The 2012 International Symposium on Applications and the Internet, SAINT 2012 (2012)
4. Fujihara, A., Miwa, H.: Effect of Traffic Volume in Real-time Disaster Evacuation Guidance using Opportunistic Communications. In: 2012 Third International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012 (2012)
5. Vahdat, A., Becker, D.: Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical report, Duke University (2000)
6. Zhang, Z.: Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. IEEE Communication Surveys Tutorials 8(1), 24–37 (2006)
7. Fujihara, A., Miwa, H.: Bluetooth & Wi-Fi mobile wireless communication experiments and the power law of passing-by frequency distributions. IEICE Tech. Rep. 110(449), IN2010-167, 139–144 (2011)
8. Fujihara, A., Ono, S., Miwa, H.: Optimal Forwarding Criterion of Utility-based Routing under Sequential Encounters for Delay Tolerant Networks. In: Third International Conference on Intelligent Networking and Collaborative Systems (INCoS) 2011, pp. 279–286 (2011)
9. Rhee, I., et al.: On the Levy-walk Nature of Human Mobility: Do Humans Walk like Monkeys? In: Proc. IEEE INFOCOM, pp. 924–932 (2008)
10. González, M.C., et al.: Understanding individual human mobility patterns. Nature 453, 779–782 (2008)
11. Spyropoulos, T., et al.: Spray and Wait: An efficient routing scheme for intermittently connected mobile networks. In: Proc. ACM SIGCOMM Workshop on Delay-tolerant Networking, pp. 252–259 (2005)
12. Lindgren, A., Doria, A., Schelén, O.: Probabilistic Routing in Intermittently Connected Networks. In: Dini, P., Lorenz, P., de Souza, J.N. (eds.) SAPIR 2004. LNCS, vol. 3126, pp. 239–254. Springer, Heidelberg (2004)
13. Burgess, J., et al.: MaxProp: Routing for vehicle-based disruption-tolerant networks. In: Proc. IEEE INFOCOM, pp. 398–408 (2006)
14. Chaintreau, A., et al.: Impact of human mobility on opportunistic forwarding algorithms. IEEE Transactions on Mobile Computing 6(6), 606–620 (2007)

15. Karagiannis, T., et al.: Power law and exponential decay of intercontact times between mobile devices. IEEE Transactions on Mobile Computing 9(10), 1377–1390 (2010)
16. Roy, R.R.: Handbook of Mobile Ad Hoc Networks for Mobility Models. Springer (2011)
17. Ano, K., Tamaki, M.: A secretary problem with uncertain employment and restricted chances. Working paper series, No. 9105, Center for Management Studies, Nanzan Univ. (1991)

Hybrid Cloud Architecture for VANET Simulations

O. Terzo, K. Goga, G. Caragnano, L. Mossucca, D. Brevi,
H.A. Cozzetti, and R. Scopigno

Abstract. The Intelligent Transportation Systems technology is meant to improve the traveling experience, for instance increasing the safety of transportation and the effectiveness of traffic management, and enhancing the environmental impact. Currently, the standardization of VANETs has reached a satisfactory stage but, despite this, only partial experimental data is available. In fact, on-field experiments are often too expensive, cannot be exhaustive (i.e., they cannot afford to test all the possible scenarios and settings), cannot be carried out in fully crowded scenarios (with many vehicles and communication systems) and will allow only limited comprehension (providing only high-level statistics).

Nonetheless, due to the complex phenomena involved in VANET (high mobility, large number of nodes and harsh environmental conditions), protocols and applications need to be extensively tested in order to guarantee reliable solutions. For this reason network simulators still play a vital role for VANETs. In fact, simulations can support both the protocol design and subsequent evaluation phases providing results and feedback under a wide range of conditions and at a lower cost than any experiment.

The objective of this work is to fill this gap through the proposal of a new architecture based on a virtual cloud computing environment for optimal scheduling of batch simulations in a hybrid cloud environment. This solution will allow improvements to be made on the performance achieved by the currently available methodologies.

Keywords: Cloud computing, Vehicular Ad-Hoc Networks, Network Simulation, Hybrid Architecture, Scheduling, IoT, Mobility.

1 Introduction

The Internet shows ever higher levels of heterogeneity (physical/real, digital and virtual, devices and device models, communication protocols, cognitive capabilities,

O. Terzo · K. Goga · G. Caragnano · L. Mossucca · D. Brevi · H.A. Cozzetti · R. Scopigno
Istituto Superiore Mario Boella, via P. C. Boggio 61, Torino, Italy
e-mail: {terzo,goga,caragnano,mossucca,brevi,cozzetti,
scopigno}@ismb.it

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 233–256.
DOI: 10.1007/978-3-642-34952-2_10

etc.) so different entities in terms of functionality, technology and application fields are expected to belong to the same communication environment. This is confirmed by the European Research Cluster on the Internet of Things (IoT), which addresses the large potential for IoT-based capabilities in Europe who have developed a vision of Future Internet [1] based on standard communication protocols where different domains are merged, namely: Internet of Media (IoM), Internet of Services (IoS) and Internet of People (IoP).

1.1 The Internet of Things and Cloud Computing

The Internet of Things is an information network made of physical and virtual things (with their own identities and attributes) seamlessly integrated into one in which communications need to take place both between people and between people and their environment. Instead, IoS denotes software components that are delivered via different networks and via Internet; IoM subtends novel approaches to share and distribute media contents, including scalable video coding and 3D video processing: media are flexibly managed and dynamically adapted to the network conditions so as to give rise to innovative applications (such as massive multi-player mobile games and digital cinema).

This future network of networks is laid out as Public and Private infrastructures dynamically extended and improved by edge points created by the things that connect one to another. Nowadays there are several types of devices comprising different types of networks that can communicate with several applications over the Internet. Network devices connect with the Internet through an interface and then communicate with applications via a backhaul network that is the communications backbone of Internet infrastructure. Communication will be seen more among terminals and data centers as in the case of Cloud Computing, than among nodes, as in current networks.

In particular, the IoT is creating a dynamic network of billions of wireless identifiable things communicating with one another and integrating the developments of concepts like Pervasive Computing, Ubiquitous Computing and Ambient Intelligence. Meanwhile, the emergence of Cloud computing has created the application and device management backbone needed to scale to and support billions of connected objects.

It was also introduced and discussed among the scientific community [1] if and how the Internet of Things could be related to cloud systems at all. The outcome could be summarized as: the Internet of Things (but also IoM, IoS, etc.) will certainly have to deal with issues related to elasticity, reliability and data management etc.; conversely, resources in cloud computing are of a type that can host and/or process data.

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2]. The cloud is made up of at least three deployment models:

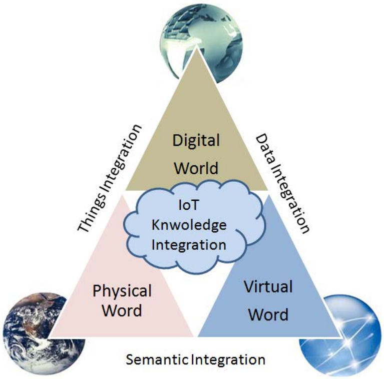


Fig. 1 Logical representation of the IoT flexibility model, as taken from [1]

1. **Public Cloud.** The infrastructure is available to the general public (or to large industry groups) according to a pay-as-you-go model. The cloud is owned by an organization selling cloud services. The advantages of public cloud service are manifolds: it is easy and inexpensive to be setup, because hardware, application and bandwidth costs are covered by the provider; it offers a scalable (i.e., progressive) approach to meet needs; resources are not wasted because you pay for what you use.
2. **Private Cloud** (also called internal cloud). The cloud infrastructure is operated exclusively for an organization. It can be a proprietary network or datacenter that uses cloud computing technologies such as virtualization[1]. The private cloud can be managed by the organization or a third party, being either on premises or off premises. It is designed to offer the same features and benefits of a cloud systems, but removes a number of objections to the cloud computing model, including worries about the possible control of corporate data, security threats, and issues connected to regulatory compliance.
3. **Hybrid Cloud** model merges two or more cloud models (private or public) that remain unique entities but are bound together by standardized or proprietary technology which enables data and application portability.

1.2 Simulation of Ubiquitous Networks Over Cloud: Topicality in the IoT Paradigm

Nowadays, cooperation among nodes has been demonstrated to enable several functions in very different environments. This is also confirmed by the mushrooming

[1] Cloud computing steps forth the idea of virtualization, which was initially circumscribed to demand paging and supervisor calls. In the last ten years, the use of virtualization in modern data centers increased on one side and to improve overall productivity by letting many more users work on it simultaneously. Even more, the cost of electricity power and the spreading green policies make virtualization more and more attractive.

paradigm of "Ubiquitous Networking", which is also addressed by ITU in the framework of Next Generation Networks (NGN) of SG11 [3]. Ubiquitous networking includes very heterogeneous media and applications spanning from Wireless Sensor Networks (WSN) to Vehicular ad-hoc Networks (VANETs), from Internet of Things to Machine-to Machine (M2M) paradigm.

The aforementioned models are very different from one to the other: in fact, they subtend different technological enabler and roadmaps (as discussed in [4] and require specific standardization efforts (as for M2M in its new Committee [5]).

Despite these differences, there are also strong commonalities among them: for instance they involve some spreading of communications so benefiting from extensive access and a cooperative approach: cooperation among cars in VANETs is to prevent accidents; cooperation among sensors in some WSNs facilitates the collection of extensive environmental information; accessibility and virtual representation of information, in IoT, also supports an increased ambient intelligence. Even more, WSNs, IoT, VANETs and M2M will have to face large numbers of nodes and, consequently, adhere to a self-organizing paradigm able to manage possible scalability issues. In fact, the self-organizing networks are networks intrinsically able to support a random number of nodes in any time of operation, with a very dynamic and reactive approach. The objects represented by IoT, for instance, would encode 50 to 100 trillion objects, and be able to follow the movement of those objects. It has been claimed that human beings in surveyed urban environments are each surrounded by 1.000 to 5.000 traceable objects [6].

However, ubiquitous networks do not present only commonalities but also interdependencies. For the sake of exemplification, in ubiquitous networks, made of billions of parallel and simultaneous events, massive parallel IT systems (parallel computing) will be required to manage the complexity of data available. This need is expected to push forward the request for new distributed computational models (such as grid and cloud computing) and for a distributed and secure way to collect, save and render available a huge amount of data: as a matter of fact, these complex structures also subtend to an interaction which can be assimilated to the IoT and M2M frameworks. In a different perspective, the study of ubiquitous networks requires computational approaches which themselves motivate the distributed computing, hence ubiquitous networking.

It is the aim of this chapter to substantiate the previous statement through the study of a particular case, involving VANETs. VANETs are an incumbent and one of the most challenging ubiquitous networks: they have been studied, so far, by means of hundreds of huge simulations, due to their safety-critical purpose (requiring extensively proven results), complex propagation environments, large number of nodes and events and, last but not least, joint mobility and communication, all of which make simulations heavier.

Even more, the validation of any new VANET feature or mechanism would require multiple new simulations. This itself hinders the deployment of new VANET solutions unless new solutions appeal for a higher scalability of the simulations. This chapter will show how distributed computing can be beneficial to the scalability of VANET simulations, providing quantitative and measurable results.

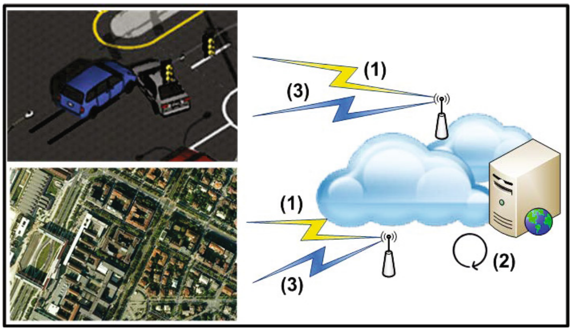


Fig. 2 An application scenario demonstrating the use of distributed computing for the real-time monitoring of VANETs and vehicular traffic, following three main logical steps: (1) Collection of vehicular information, (2) Traffic-routing decisions based on real-time simulations, (3) Distribution of traffic management information through the VANET

Notably, the approach here presented can be extended, in future, beyond the case of modeling by simulations, so as to also cover a real-time processing. Suppose that one needs to make decisions for the routing of vehicular traffic and to inject these policies leveraging the existing VANET protocols. In this case the following logical steps should be covered:

1. Collection of vehicular information (through a WSN, a VANET or other ubiquitous networks);
2. Traffic-routing decisions based on real-time simulations, necessarily over distributed resources, for the sake of scalability. This step could include also the simulation of policy efficacy, considering existing VANET protocols;
3. Distribution of traffic management information through the VANET;
4. Continuous monitoring of the effectiveness of traffic routing and feedback onto the provisioned policies.

The steps from 1 to 3 are also depicted in the example of Fig. 2.

This scenario makes the mutual connections among different types of collective intelligences over IoT ever more complex: vehicular traffic may be represented and monitored according to an IoT paradigm, continuously processed over distributed intelligence and actuated through the cooperative approach enabled by VANET communications.

The potentially critical role of cloud computing emerges ever more dramatically in the case of VANETs: in fact, an overall multi-layer VANET simulation should collect and mutually coordinate the results from different simulation platforms which distinctively address specific phenomena (e.g., propagation, networking, mobility, etc.).

An example would be where results were presented in a novel paradigm for combining different simulators [7]. The approach is based on the IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA).

There are also further possible points of contact between VANET and cloud computing: while they are not covered in this chapter, a brief discussion is here presented for the sake of clarity.

The challenge revolving around these possibilities include cloud architecture being built by the vehicles themselves, the creation of car clouds, and also the enabling of new services [8]. For example, a company's parking lot will embody a sink of computational resources and/or storage and could be used, as a classic public cloud infrastructure - given a certain rewarding policy. In this case a novel perspective is provided, quite converse to the initial statements: cars do not use cloud potential but could even build a cloud. Interestingly, these scenarios pose new challenges that should be faced in the future: for example nodes could leave the network (due to mobility) or vary their computational resources made available to the cloud, depending on the state of the vehicle. As mentioned before, this topic is beyond the scope of this chapter and it can represent the next generation technology in VANET research.

The remainder of this chapter is structured as follows. First the introduction is completed by a discussion on the topic of cloud computing in the general framework of IoT. Section 2 is devoted to the introduction of the topic of VANET in general and, then, of VANET simulations, primarily focusing on the aspects connected to scalability. Section 3 describes cloud computing and, in more detail, the so-called elasticity model. Section 4 puts things together and shows a practical case in which a hybrid cloud architecture is leveraged to carry out several heavy VANET simulation tasks. The aim of the proposed solution is to show the most effective distribution of the designated tasks, thus Section 5 is meant to present the performance evaluation and lastly, conclusions are drawn in Section 6.

2 VANET Networks: Characteristics and Scalability Issues

A Vehicular Ad-hoc Network is composed of vehicles exchanging data on a wireless channel, internationally set in the 5.9 GHz range [9]. Due to the wireless connection and mobility, VANETs rely on temporary links that carry the communications between nodes. In a simplified taxonomy VANET communications can be classified into V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure, that is a network of fixed VANET nodes) although other classes have been recently proposed.

Currently, the infrastructure is forecast by the standards, but is not yet mandatory in VANETs: this is to prevent the need for an additional prohibitive cost as a prerequisite for the deployment of VANET. On the other hand, this means that the network must be able to work in a completely distributed manner and without a centralized coordinator: the main effect is on the medium access control protocol (or MAC the protocol coordinating transmissions in the wireless medium), which must be able to work in absence of any central coordinator, that is, in a decentralized (fully distributed) way.

As already said, the main purpose of VANETs is the enforcement of safety; however, to encourage the deployment, also non-safety, possibly commercially distributed, services are planned (e.g., for traffic improvement and drivers comfort).

Given the primary safety purpose, the reliability and security of data exchange inside VANETs must be carefully addressed.

Even more, the standardized solutions by IEEE (IEEE 802.llp [10]) and ETSI (ETSI EN302571 [11]) are based on the well-known IEEE 802.11 (WiFi) standard. In fact, IEEE 802.11 includes the most common decentralized MAC protocol, which is based on the CSMA/CA algorithm: each station senses the medium before transmitting (Carrier Sensing) and prevents simultaneous transmission through a statistical waiting time (*Collision Avoidance*).

CSMA/CA is simple and well-known, however the Scientific Community has some major concerns about its flexibility capabilities to support timely and reliable delivery of real-time messages, as required by safety oriented applications, especially under congestion and in presence of obstructions (hence *hidden terminals*).

This constitutes a major threat to the extent that two counteractions are being studied: the Specialist Task Force STF395 of ETSI has investigated both *(i)* congestion-control algorithms [12] and *(ii)* distributed, connection/oriented, synchronous slotted protocols [13, 14]. Among the slotted protocols MS-Aloha [15, 16] has been demonstrated, through simulations, to potentially improve communications both under congestion and with hidden terminals.

Overall, VANETs, despite being standardized, still require studies to overcome their current limitations. Despite the availability of IEEE 802.11p transceivers, these analyses are carried out primarily by simulations: on-field experiments are often too expensive, cannot be exhaustive (i.e., they cannot afford to test all the possible scenarios and settings), cannot be carried out in fully crowded scenarios (with many vehicles and communication systems) and allow only a limited comprehension (providing only high-level statistics).

Simulations will serve not only the purpose of improving protocols, but also the validation and consecutive introduction of new services based on VANET communications. For example, some new paradigms are being discussed, such as the V2G (Vehicle to Grid) for joint management of the green routing of cars, of the charging state of the vehicles and the availability of charge on the energy-grid. Similarly, V2C (Vehicle to Cloud), could be the solution for traffic management, as preliminary discussed in Section 1.2. With V2C, information related to the traffic, to the pollution, etc., could be collected and elaborated in real time in order to give feedback to the drivers, with the VANET becoming an extension of the Internet of Things, connected to a cloud. These scenarios will, of course, require a preliminary validation by simulations, too.

Having said that however, VANET simulations still play a vital role. In the following subsection the issues connected to them are now introduced.

2.1 Scalability of Simulations

The increasing attention of the scientific community in regards to VANETs has encouraged researchers to study and develop more and more accurate and realistic simulation tools. There is a lot of software used in VANET research [17, 18].

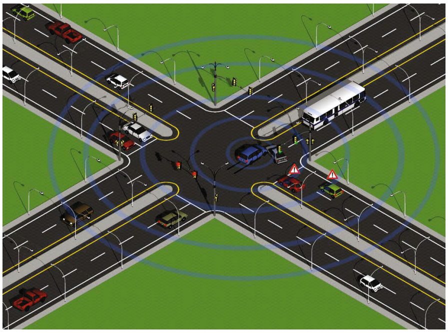


Fig. 3 A well known VANET scenario: safety enforcement through wireless communications between cars approaching a crossroad

Network simulation has been demonstrated to lead to results which are close to what is found in practice [19]: this is a notable result achieved through a long lasting process of simulator improvement. Over the last few years more and more features have been added, especially to the open-source simulators thanks to the contributions from a large scientific community: attenuation and obstruction models, fading description, and more detailed analyses of the phenomena inside the receiver, for instance, have increased the realism of simulations.

Such richness in detail involves heavy computational loads: thus, the lack of support for multiprocessing by network simulators may become a critical weakness. For example both softwares NS-2 [20] and NS-3 [21] (the well-known and most used network simulator for vehicular networks) are monolithic programs which cannot exploit the presence of a multicore to speed up the simulations. An exhaustive survey and comparative study of these tools can be found in [22].

The lack of parallelism in network simulators could become even more critical in the future. Firstly, concerning the number of vehicles, todays simulations consider hundreds of nodes but, for real complex urban scenarios, thousands of nodes should be supported. Even further, considering mobility, node positioning should change over time and possibly, also change depending on protocol exchanges (VANET-driven rerouting of vehicular traffic).

Another aspect that heavily affects the simulations is the wireless channel model. Usually the wireless channel is modeled by simple fading models (i.e., Nakagami) and often the scenario is approximated as free space within an urban environment, as well. A more realistic simulation should involve ray tracing techniques to include the effects of obstructions and scatterers from the real world. Radio Frequency (RF) simulations are very heavy and scarcely computationally scalable. The authors have proposed a method [23] to summarize results from ray tracing simulations so as to integrate them within NS-2: the method, called RADII [2], subtends the computation of all possible paths and the resulting received power between each couple of points on a map.

[2] RADII: RAy-tracing Data Interpolation and Interfacing.

The process is useful to minimize the load of the simulation but with the drawback of adding an approximation in the calculus and involving some manual steps (not automatically computed). As a result, if computational loads were not a problem, the direct ray-tracing simulations could be preferred to the models created by RADII. Furthermore, the richness of wireless models could be further increased to include the characterization of antenna behavior (i.e., the actual antenna radiation pattern) and Multiple Input Multiple Output (MIMO) techniques.

Simulations could be made more realistic by also taking into account the other vehicles surrounding each transmitter and receiver. Typically, simulators manage vehicle as entities without dimensions which do not influence signal propagation. In its current state of art, some early experimental works addressed this problem but the impact of this aspect has been largely neglected in simulations although it is expected to be relevant and computationally heavy [24].

In the near future, simulations should take in account further additional information. For instance, upper layer protocols (e.g., CAM messages [25] regarding the status of the car) could be managed. This intelligence, in principle, will allow an evaluation regarding safety effectiveness in specific scenarios, giving a complete, top-down characterization. This will increase the computational load too.

Overall parallel computing appears very critical for the sustainability of future network simulations. It will require implementation of new network simulators, so as to enable parallel computing itself.

However, parallel processing can be leveraged also by monolithic programs. This is the case when several simulations need to be carried out in order to understand how a certain parameter (e.g., transmitted power, priority of messages, number of nodes) affects the quality of transmissions. In this case the simulations could potentially benefit from spreading over a network of computational resources throughout the cloud. This however requires a deeper understanding of the parameters which may affect the complexity of VANET simulations and allow the best allocation of cloud resources to each simulation.

3 Cloud Computing: Elasticity Model

Cloud Computing fortifies some concepts on how to build highly scalable Internet architectures and presents new aspects that revolutionize the way in which applications are deployed. The cloud has introduced changes in several processes, practices, philosophies and fed up service-oriented architectural principles that industry experts have deemed important [26]. Traditional (*non cloud-ready*) applications were built adhering to models and philosophies which cloud has surpassed, as briefly discussed previously.

3.1 Design of Scalable Cloud Architectures

A Hybrid Cloud infrastructure is designed to provide conceptually infinite scalability [27]. However, it is not possible to leverage all that scalability available in the infrastructure if the architecture is not scalable.

So, when designing a cloud architecture, a critical analysis must address the identification of the monolithic components and the bottlenecks within: this is to be aware of the areas where it is possible to exploit the on-demand provisioning capabilities; the main application can be better adapted to a scalable paradigm, so as to best benefit from the cloud.

There are different characteristics of an efficient scalable application:

- increasing resources result in a proportional improvement in performance;
- heterogeneity;
- efficiently;
- resilience;
- the more it grows, the more cost effective it should become (per-unit cost reduces as the number of units increases).

These aspects should become an implicit part of a cloud application and IT specialists should keep it in mind when design new architecture: architecture and infrastructure will be mutually synergic and lead to the expected scalability.

3.2 Elasticity and Scalability

Fig. 4 shows different approaches which a cloud architect can consider when delivering a solution which matches the scalability demanded by the application.

The first approach is called *scale-up* and it is not concerned by scalable application architecture; it is possible to invest heavily in larger and more powerful computers (vertical scaling) to accommodate the demand. This approach usually works but up to a point: it could either cost a fortune (Large capital expenditure in the diagram) or the demand could exceed the capacity before the new big iron is deployed (see “insufficient hardware” in Figure 4).

Instead the traditional *scale-out* approach creates an architecture that scales horizontally so that the investments in the infrastructure are in small chunks. Most large-scale applications follow this approach: they distribute the components (of the application), federate their datasets and employ a service-oriented design. This approach is often more effective than a scale-up approach. However, this still involves predicting the demand at regular intervals and subsequently deploying an infrastructure in chunks to meet the demand. This often leads to excess capacity and constant manual monitoring.

A non-cloud infrastructure generally needs to predict the amount of computing resources your application will use over a period of several years. If you underestimate, your applications will not have the horsepower to handle unexpected traffic, potentially resulting in customer dissatisfaction. If you over-estimate, you will waste money with superfluous resources. The on-demand and elastic nature of the cloud approach (*automated elasticity*), however, enables the infrastructure to be closely aligned (as it expands and contracts) with the actual demand, thereby increasing the overall utilization and reducing cost.

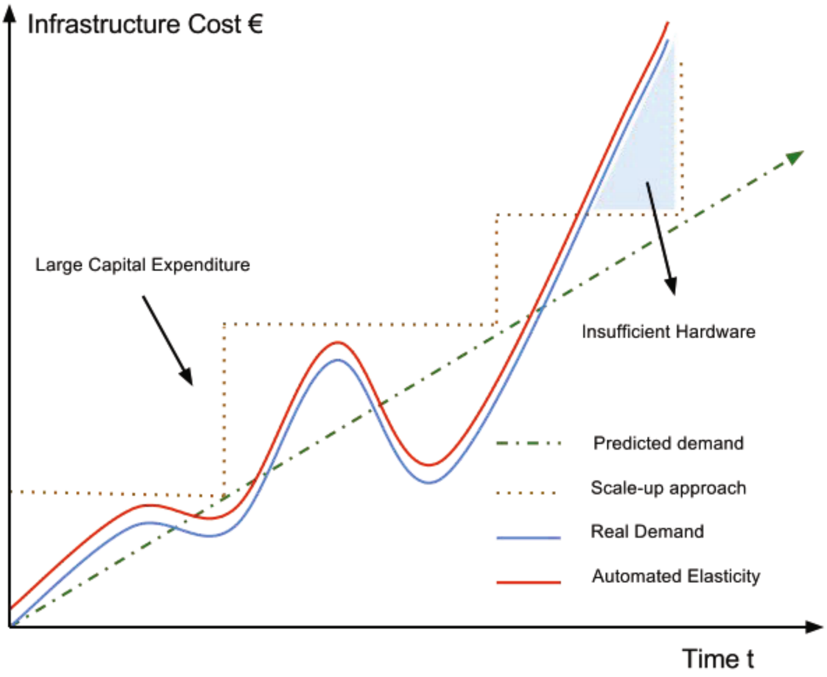


Fig. 4 Elasticity and scalability in cloud

Elasticity is one of the fundamental properties of the cloud: it allows to scaling of computing resources up and down easily and with minimal friction. It is important to understand that elasticity will ultimately drive most of the benefits of the cloud.

4 VANET Simulations Over Cloud

This section presents the practical case of VANET simulations over a hybrid cloud infrastructure.

For the sake of clarity, before implementing the cloud infrastructure, tests were made in a traditional system. The objective of such tests is to *learn* how to estimate the execution time of simulations and system performance when working in a non-cloud (Section 4.2). The tests will drive the sizing of the cloud infrastructure whose technical characteristics are presented in the following paragraphs (Section 4.3–4.4).

This section also presents in 4.5 the scheduling core which makes it possible to properly distribute simulation tasks on a hybrid cloud platform.

4.1 Learning Phase

The first key step, called *learning phase*, is meant to enrich the “knowledge” of the computational resources required by each simulation, depending on its main settings.

Notably, learning phase is carried out only once, at the beginning, when the whole architecture is implemented. Afterwards, when the service is active, the system will auto-learn by each task, updating the estimation of simulations. This phase is essential for the optimization of the overall simulation process. In fact, the scheduling core of the architecture leverages the information to distribute simultaneous works among several hardwares.

The learning phase involves a preliminary definition of the main metrics addressing the computational load. Based on the authors' experience, the following three *simulation metrics* (SM) have been considered:

1. amount of allocated memory (RAM);
2. simulation time;
3. size of the output file.

Different combinations of these parameters are expected to produce a different impact on the overall computational load. In order to derive a precise analysis of the used resources, a simulation set is tested and each of them is performed sequentially (without any other process in parallel) over a single machine (worker node).

In all the simulated VANET scenarios, the number of vehicles is fixed throughout the simulation, their mobility patterns are *a priori* known, all the mobile nodes generate packets at a same rate and the transmitted power is the same across the network. Consequently, the following high-level parameters have been considered to assess the SMs:

1. number of vehicles;
2. simulated (elapsed) time (sec);
3. application rate - packet/sec (Hz);
4. transmitted power (dBm).

A large set of simulations (more than 100) have been carried out to accomplish the learning phase. The following settings have been adopted:

- number of vehicles [50, 100, 250, 500, 750, 1000];
- simulation time [40, 100] sec;
- application rate [5, 10, 15] Hz;
- transmitted power [7, 10] dBm.

Through this, it can be supposed to achieve a satisfactory initial knowledge about the computational demand subtended by simulations of vehicular networks. The outputs of the simulations have been measured by the SMs. These raw results have been rounded up into discrete values to facilitate the measurement and the scheduling operations; conversely, the approximation is considered negligible with respect to the implicit statistical variations in the measurements, and negligible also with respect of other hardly measurable dependencies (on the operating system, on the used hardware, etc).

The values resulting from the learning phase constitute a worst-case starting point which is refined while the following jobs will run. Table 1 shows the resulting raw values: they facilitate the assignment of appropriate weights to simulations for the task composition.

Table 1 Some simulation results of the learning-step

Sim	#Vehicles	Simulation Time [sec.]	Tx Power [dBm]	Pkt rate [Hz]	Elapsed time [hh:mm:ss]	RAM [Mbyte]	Output size [Mbyte]
Sim 1	1000	100	10	15	03:39:20	409.9	24400
Sim 2	1000	100	10	10	03:27:53	386.2	23900
Sim 3	1000	40	7	10	03:40:53	181.8	23990
Sim 4	1000	40	7	5	02:06:37	179.8	12364
Sim 5	750	100	10	15	03:39:07	150.1	23900
Sim 6	750	100	10	10	03:38:13	147.4	23307
Sim 7	750	40	7	10	01:57:09	141.8	14785
Sim 8	750	40	7	5	01:07:46	135.5	7400

4.2 Proposed Architecture

This subsection is meant to present and substantiate the decisions made in designing the hybrid cloud architecture which was adopted for VANET simulations. Concepts like “elasticity” and “virtual environment” have already been introduced: they have emerged, respectively, due to dynamic nature of cloud and to move a scientific-class application from a fixed physical environment to a virtualized cloud environment. Even more, the concept of hybrid architecture [2] on a cloud computing environment can be recalled: in the authors’ experience, resulted as being particularly beneficial in improving simulation performance.

Indeed, to perform many VANET simulations, a large amount of computing power is required, but if you have to build a large scale system it may be too expensive: one should invest in hardware (server, routers, power supply, racks), power management, cooling and personnel costs.

Cloud computing renders IT infrastructures easier to manage, and, at the same time, provides high performance, massive scalability and reliability. However, a pure private cloud environment cannot always provide all the resources required by an application without considering an unpredictable growth. As an application grows in popularity and complexity, the available resources of the private cloud may no longer be sufficient and as a result, many organizations consider a hybrid cloud solution which allows strategic access to both the private and public cloud resource pools. A hybrid cloud architecture opens the application to the infinite resources of the public cloud.

A hybrid architecture fits well to the needs of researchers addressing the problem of performing simulations or very intensive tests for a short period of time, without needing to buy additional hardware. Not making such an investment would mean having to avoid hardware that might be unused most of the time. For this study, a cross-platform architecture was used: it consists of interconnected Virtual Machines. The system is developed according to a hybrid cloud computing model [2]. According to the NIST definition, a hybrid cloud is a combination of a public and private cloud that interoperates. In this model users typically outsource non-business-critical information and processing to the public cloud, while keeping business-critical services and data in their control. In this work the hybrid architecture comprises two systems: the first consists of VMs belonging to a private cloud platform, the second is a set of VMs available from a public cloud platform (for our experiments we used Amazon EC2). Both private and public cloud are fully integrated and managed as a whole environment able to perform a large number of simulations in parallel.

In regards to the simulation platform, a network simulator tool NS-2 has been chosen as it is widely accepted within the scientific community. However – this is very important – any other network simulator cold be used.

4.3 Overview of the Cloud Infrastructure

Cloud Computing has gained a lot of popularity in recent years. Cloud customers outsource their computation and storage to public providers and pay for their service *on demand*. Cloud providers offer a highly reliable and scalable infrastructure for deploying web-scale solutions, with minimal support and administrator costs, and made flexibility.

For this study's hybrid architecture, authors adopted a cloud computing service hosted in Amazon Elastic Compute Cloud (EC2). Amazon EC2 is an IaaS (Infrastructure as a Service)[2] and it is part of Amazon's cloud computing platform, Amazon Web Services (AWS). EC2 was chosen because it is currently a feature-rich, stable and commercial public cloud. It also offers a web service through which users can boot an Amazon Machine Image (AMI)[28] to create an "instance" (virtual machine) and rent virtual machines. Amazon EC2 is based on the XEN para virtualization technology [29] and it is possible, as well, to size instances based on EC2 Compute Unit (ECU) [30].

Amazon adopts virtualization technology that allows you the possibility of a flexible configuration with the characteristics of a resource instance. Therefore, the features of the virtual machine (such as the number of CPU cores, the processing power per core, memory, performance I / O, etc.) are fully user-configurable parameters and a server can be chosen from a series of configurations. Even the operating system and the software that runs on the virtual server can be completely customized by the user (restrictions may apply to the OS kernel). One of the most interesting feature is that user can create an image of the software configuration that can be used on multiple servers.

Amazon offers pay-per-use services, so that resources are paid based on: the processing capacity of each instance, the size of storage requests (GB) and network traffic generated [30]. Concerning the addressing, elastic IP addresses are used: they are static IP (IPv4) designed for dynamic cloud computing; they belong to the account and not to a virtual machine instance and exist until it is explicitly released by the user [30].

For the sake of completeness, there are several providers that offer almost the same services as Amazon EC2. Amazon EC2 was used for this study because it offers the best price/value ratio [31], in the authors' perception. Currently, compared to other platforms (e.g. GoGrid) Amazon EC2 has a lower time of resource allocation and release that affects the costs and generally the Amazon EC2 instances have better performance compared with similar systems, for the sequential operations (typical to scientific computing) [31].

4.4 Public Cloud Platform with Amazon EC2

A hybrid cloud necessarily needs to connect two different facilities: a private and a public cloud. The hybrid cloud architecture (see Fig. 5) consists of a master node and three worker nodes on the private cloud and one worker node on the public cloud (Amazon EC2). Nodes are virtual machines with a linux operating system and all the required services. The private cloud virtual machines (guests) reside on two different hosts (connected to a 100 Mbps Ethernet network). Virtual machines use bridged networking, so that virtual interfaces are used to connect to the external network through the physical interface, appearing as regular hosts to the network.

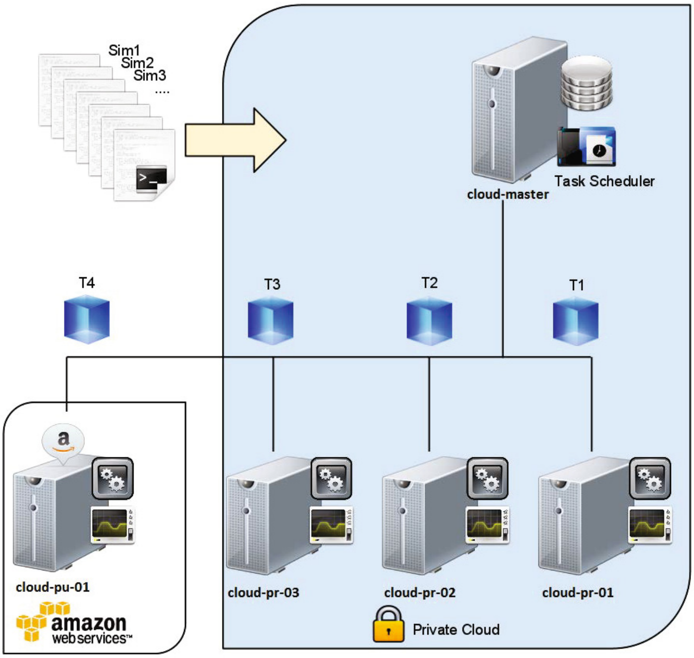


Fig. 5 Hybrid cloud architecture for VANETs

Hardware configuration of nodes is very similar for processors, and disks but different in the amount of memory (RAM) allocated. In a private cloud, nodes are fully virtualized [29] and use two virtualization solutions:

- XEN - The Xen virtualization platform is an open source software and it is one of the few hypervisors that supports both para virtualization and full virtualization.

Xen Hypervisor is the direct interface between guest virtual machines and the hosts hardware, and receives all requests for CPU, I/O and disk usage. Due to the separation between the OS and hardware, the hypervisor can run multiple operating systems safely and concurrently.
- KVM - Kernel-based Virtual Machine is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, kvm-intel.ko or kvm-amd.ko.

In addition, master node hosts a database that contains all the information about *simulation weights*, *machines configuration* and *virtual machines states*. At each simulation file, a weight is assigned according to the required RAM memory, read from simulation weight table; the table on machine characteristics is updated with the data about the free memory (RAM), processor speed (CPU) and available disk space of each virtual machine. Further, the table on virtual machines state stores information about availability of VMs.

The *Task Scheduler* resides on the master node, which is an application developed in java language, building tasks (one or more simulation files to be executed) based on the free memory of the worker nodes and sends them to the available VMs. In each worker node an agent and a system monitor runs. Periodically, the agent detects whether there are tasks to be performed and executes them. The agent also sends information about the status of the worker nodes (active, inactive) and about the system load (e.g. free memory) to the database.

4.5 Scheduling Algorithms

When the cloud platform is fully operational, each worker node is ready to run its simulations. The user prepares a set of simulations that must be performed and transfers them to the master node. The simulations are described and detailed in OTcl language scripts. The process starts when the master node receives a subset of simulation (Object Tcl files) to be executed. A Task consist of a set of *n* simulations that the master node has to distribute among the worker nodes in order to perform them in parallel. To prevent resources' overload, worker nodes compute a task at a time. According to the initial analysis, it is possible to detect the weight of each simulation file in terms of used resources. In fact, a relation is inferred between the simulation files and the allocated memory (RAM) (see Section 4.1).

Through a scheduling algorithm, a master node builds a task tailored to the available memory (RAM). In this way, each node executes a task that allocates only the RAM required to run the whole process. Scheduling is performed by a task scheduler which is a software module installed on the master node: it can be considered the core of the cloud infrastructure. The scheduler continuously reads information from the database to check the availability of the nodes and the queue of simulation files waiting for processing.

The scheduler builds tasks through three steps: detecting available nodes, acquiring memory info and selecting simulation files. In the first step, the master node queries the database to know how many and which worker nodes are in *available* state. When a worker node changes state from *available* to *running*, it will receive a task to execute. Instead, when a worker node changes from *running* to *available*, it has completed the execution of the task and can accept a new one.

During the second step, task scheduler queries database to obtain the exact value of free memory of each worker node.

In step three, after compiling a list of available worker nodes, the scheduler selects a subset of simulations from the queue and collects them in a single new task. Finally, a task is delivered to each worker node.

The task builder is an algorithm that assembles a set of simuations (OTcl files) according to the criterion described below. First of all, OTcl queue file is identified; then the relevant OTcl files are selected so that the sum of the weights (memory allocated by a simulation) does not exceed 90% of the total available memory on the worker node. As an additional rule of thumb, the maximum number of simulation files to be executed in parallel should not be greater than 4, so as to prevent the CPU overloading. The task created is sent to the respective worker node and then the agent runs all the simulation files in parallel.

When the processing is concluded, the agent reports to the database that the worker node is ready to accept new tasks. The algorithm will continue from the beginning until the queue of files is emptied.

5 Performance Analysis

The first practical result were obtained during the learning phase (simulation time, size of the output file and amount of allocated memory (RAM)) and were used also for the initial design of the cloud platform. In particular, the weight of each simulation mainly depends on the memory (RAM) required by the simulation. The weight of each simulation is used by the task scheduler to create the appropriate tasks and spread them among worker nodes. Size of the output files were used in order to create virtual machines with hard disks of the appropriate size. The simulation time obtained during the learning phase was compared with simulation time on the hybrid cloud platform in order to amend the values.

The hybrid cloud platform is composed of five virtual machines, four of them reside on the private cloud and one on the public cloud (Amazon EC2). The characteristics of the virtual machines are shown in Table 2.

The Amazon EC2 instance is an EBS [32] Standard Large Instance (m1.large) with the following characteristics [30]:

- 7,5 GB of memory;
- 4 EC2 Compute Units[3] (2 virtual cores with 2 EC2 Compute Units each);
- 850 GB of instance storage;
- 64 bit platform;
- cost $0.36 per hour.

[3] One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

Table 2 Virtual machines characteristics

VMs	OS	CPU model	Virtual CPU	RAM (GB)	Kernel
cloud-master	Debian Squeeze 6.0.2	Intel Xeon 5140 @ 2.33 GHz	2	1	Linux 2.6.32-5-xen-amd64
cloud-pr-01	Debian Squeeze 6.0.3	Intel Xeon 5140 @ 2.33 GHz	2	3	Linux 2.6.32-5-xen-amd64
cloud-pr-02	Debian Squeeze 6.0.3	Intel Xeon 5140 @ 2.33 GHz	2	2	Linux 2.6.32-5-xen-amd64
cloud-pr-03	Debian Squeeze 6.0.3	Intel Xeon X5660 @ 2.80 GHz	2	7.5	Linux 2.6.32-5-amd64
cloud-pu-01	Debian Squeeze 6.0.1	Intel Xeon E5507 @ 2.27 GHz	2	7.5	Linux 2.6.32-5-xen-amd64

For the performance analysis 46 simulations were executed, while three types of test assessed the effectiveness of the scheduling algorithm, resource checks and memory usage.

5.1 Effectiveness of the Scheduling Algorithm

The aim of this test is to demonstrate that scheduling algorithm improves performance if compared to the sequential execution of simulation files. In this test, the algorithm assigns tasks to worker nodes considering only the available memory (RAM) according to the condition:

$$W_t = k \times M_n \tag{1}$$

where:

W_t is the weight of the task to be executed by the worker node;

$$W_t = (\mu_1 + \mu_2 + \mu_3 + \ldots + \mu_n) \tag{2}$$

μ_n is the weight of a simulation file;

k - is a multiplier coefficient; M_n is the allocated memory of the worker node.

In this test k=1 so the Task Scheduler assigns the tasks according the condition:

$$W_t = M_n \tag{3}$$

In order to accomplish this test 46 simulation files were divided into 20 tasks. In Table 3 the following data is shown: tasks executed on the virtual machine, number of simulation files, execution time using the task scheduler and execution time of simulation files performed sequentially.

The values of this table are represented also in Fig. 6: the use of the scheduling algorithm decreases the simulation time for those tasks made of multiple simulation files (Task 1, 3, 4) and it is almost the same for the tasks made up by a single simulation file (Task 2, 5, 6).

Table 3 Simulation time on cloud-pr-03

Task	Simulation files	Time (hh:mm:ss)	Time_seq (hh:mm:ss)
Task 1	4	01:22:44	1:54:25
Task 2	1	01:29:09	1:30:33
Task 3	2	00:52:28	1:42:24
Task 4	2	00:44:36	1:22:10
Task 5	1	00:34:05	0:34:21
Task 6	1	00:33:10	0:33:28

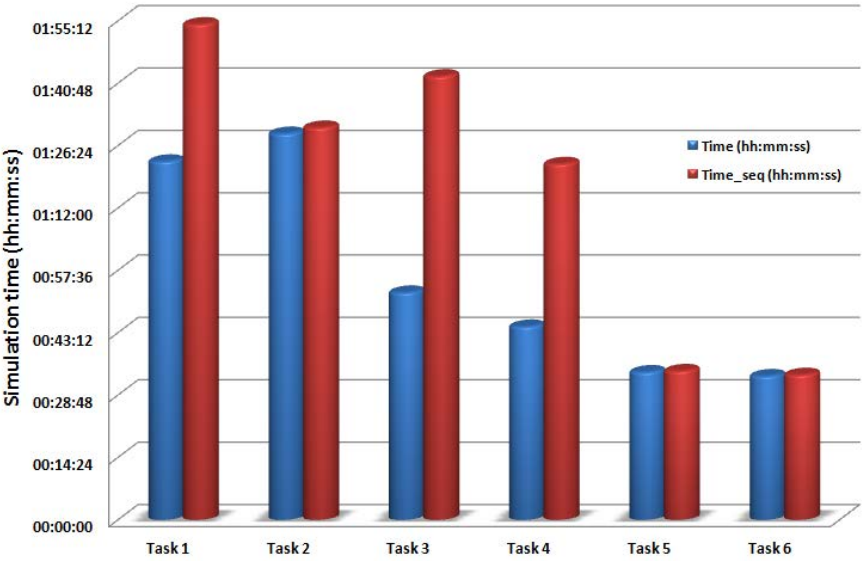


Fig. 6 Simulation time on cloud-pr-03

Notably, the latter tasks are generated when the weight of a simulation file μ_1 almost equals the available memory M_n of the worker node $\mu_1 \approx M_n$ and the task scheduler is not able to add any other simulation file to the task, since no simulation file meet the condition:

$$\mu_2 < M_n - \mu_1 \tag{4}$$

The total simulation times of all the tasks executed over the hybrid cloud platform are shown in Table 4. The most powerful virtual machines are cloud-pr-03 and cloud-pu-01: they are respectively assigned 6 tasks (11 simulation files) and 2 tasks (9 simulation files). This happens because the task scheduler assigns preference to the most powerful machines the tasks containing grater weight simulation files (that is requiring more available memory and computation time).

Table 4 Total simulation time using the scheduling algorithm

VMs	Tasks executed	Simulation files	Time(hh:mm:ss)
cloud-pr-01	3	10	00:59:30
cloud-pr-02	9	16	00:52:30
cloud-pr-03	6	11	05:36:12
cloud-pu-01	2	9	01:39:37

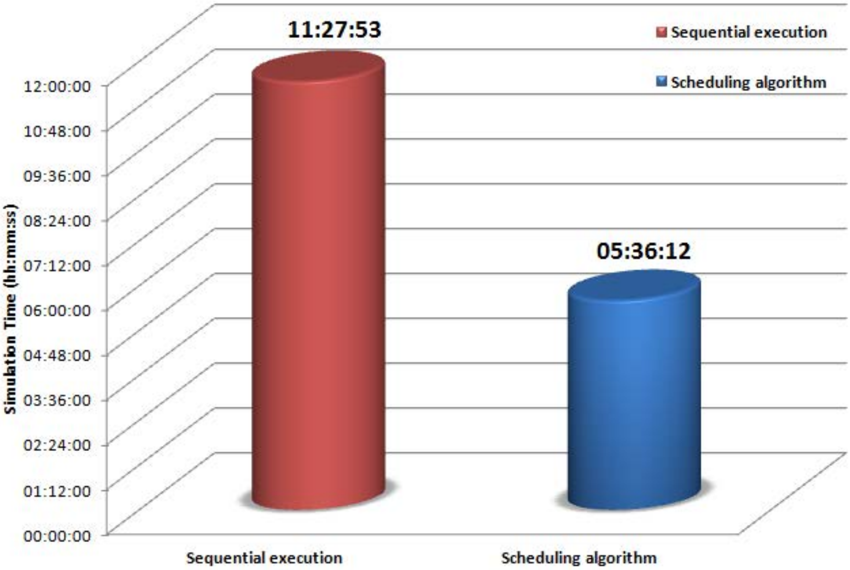


Fig. 7 Total simulation time sequential execution vs. scheduling algorithm

In Fig.7 the total simulation time of the validation algorithm is compared to the total simulation time of the sequential execution. It can be noted that the use of the Task Scheduler reduces the total simulation time significantly (by almost 48%).

5.2 Scheduling Algorithm Optimization - Resource Usage

During the preliminary phase of the validation, it was found that smaller virtual machines (see Table 4) remained in *available* state for long periods of time – without performing tasks – because only simulations exceeding weights were available. Thus the scheduling algorithm was modified in order to optimize the resource usage, considering the free memory (RAM) of the worker nodes: if there is a worker node in an available state but without enough free memory to execute the simulation files of the queue, it is assigned a task (consisting of a single simulation file) to be performed. This means 46 simulation files, divided into 23 tasks, have been executed – as shown in the Table 5.

Table 5 Total simulation time using the optimized algorithm

VMs	Tasks executed	Simulation files	Time(hh:mm:ss)
cloud-pr-01	5	11	03:20:15
cloud-pr-02	10	17	03:24:23
cloud-pr-03	4	10	02:32:02
cloud-pu-01	4	8	03:17:36

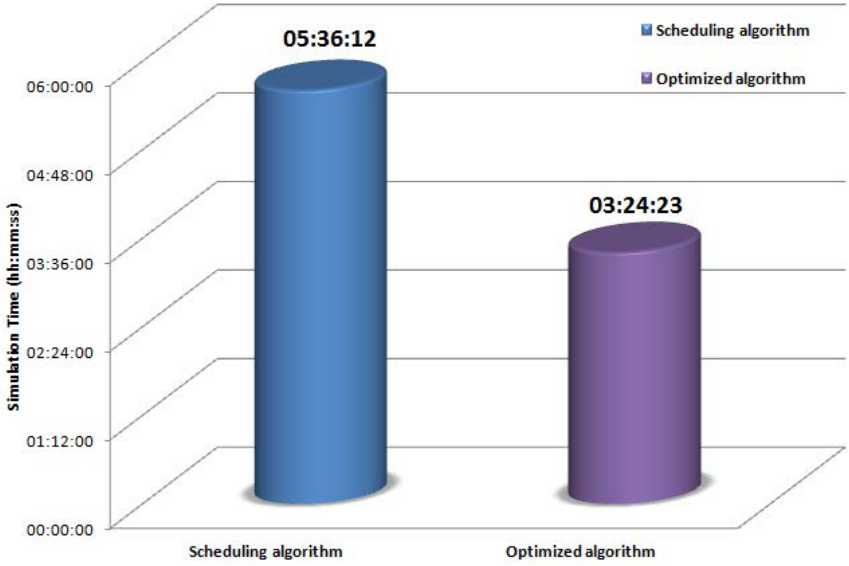


Fig. 8 Total simulation time scheduling algorithm vs. optimized algorithm

Altogether, comparing the results obtained and plotted in the Fig.8, the optimized algorithm reduces the simulation time considerably.

5.3 Scheduling Algorithm Optimization - Memory Usage

During first two tests on the effectiveness of the algorithm and optimization of the resources, the parameter k was set to the value 1 (see Eq. 1), so that each virtual machine was assigned a task equal to its allocated memory (RAM). However, to understand how the simulation time changes as a function of the coefficient k, the same 46 simulation files were executed using the optimized algorithm, but with changes to the coefficient $k = 1.5, 2, 2.5, 3$. Results are shown in Fig. 9 the hybrid cloud platform performance improves up to k = 2, while beyond performance worsens.

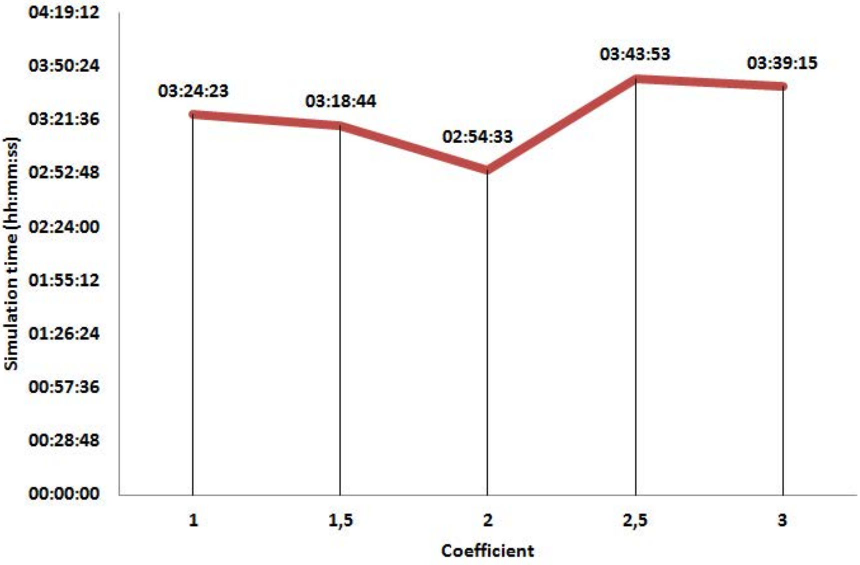


Fig. 9 Total simulation time for different *k* coefficients

6 Conclusions

Internet of Things paradigm is creating a network of billions of wireless identifiable things communicating with one another and integrating developments from concepts like Pervasive Computing, Ubiquitous Computing and Ambient Intelligence. These are increasing the amount of data to be elaborated and stored considerably : system based on Cloud computing can help these needs.

VANETs represent a particular type of ubiquitous network of the future. VANETs not only will require cloud computing for their management in the future, but also leverage cloud capabilities today for the purpose of simulations. Indeed a solution is here proposed, based on a novel Hybrid Cloud Computing platform, and aimed to improve performance of the heavy computation required by VANET simulations.

The adopted infrastructure is a cross-platform cloud architecture made up of virtual machines which allow the running of a large number of simulations in parallel in order to reduce the total simulation time. The analyzed simulation software (NS-2) could not exploit the benefit of a multiprocessor environment which means that the number of performed simulations is tied to the number of machine processors. The Head of architecture is a scheduler that is used to assign simulations to VMs after merging them into tasks, depending on available resources (memory) on each node.

From a research perspective, the impact seems promising. In fact, results demonstrate that a deep resource optimization is achieved, while execution time and costs of simulations is significantly reduced by acting on the available memory (RAM) in each node.

The simulation time is reduced by almost 48% by a scheduler that assigns the various tasks to the virtual machines optimizing certain parameters.

For the authors now, the main issue which remains open concerns the manual work necessary to arrange the simulation files: the automation of this process could make the overall scheme even more adaptable and this will be part of the authors' future work.

References

1. Vermesan, O., et al.: Internet of Things Strategic Research Roadmap (2012), http://internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf
2. Mell, P., Grance, T.: The NIST definition of Cloud Computing (2012), http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
3. International Telecommunication Union, http://www.itu.int
4. Ashton, K.: Internet of Things Strategic Research Roadmap. CERP-IoT, September 15 (2009)
5. Scarrone, E., Boswarthick, D.: Overview of ETSI TC M2M Activities (March 2012), http://docbox.etsi.org/M2M/Open/Information/M2M_presentation.pdf
6. Waldner, J.B.: Nanoinformatique et intelligence ambiante. Inventer l'Ordinateur du XXIeme Siècle, p. 254. Hermes Science, London (2007) ISBN 2746215160
7. Schuenemann, B.: V2X Simulation Runtime Infrastructure VSimRTI: An Assessment Tool to Design Smart Traffic Management Systems. Computer Networks 55(14), 3189–3198 (2011) ISSN: 1389-1286
8. Abuelela, M., Olariu, S.: Taking VANET to the Clouds. In: The 8th International Conference on Advances in Mobile Computing & Multimedia, MoMM 2010 (2010)
9. Hartenstein, H., Laberteaux, K.P.: A tutorial survey on vehicular ad hoc networks. Proc. of the IEEE Communications Magazine 46(6), 164–171 (2008)
10. IEEE Standard 802.11p, Wireless Access in Vehicular Environments (July 2010)
11. ETSI EN 302 571 - Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive, http://www.etsi.org/deliver/etsi_en/302500_302599/302571/01.01.01_60/en_302571v010101p.pdf
12. ETSI TS 102 687 Intelligent Transport Systems (ITS); Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range. Access layer part (2011)
13. ETSI TR 102 862 Intelligent Transport Systems (ITS); Performance Evaluation of Self-Organizing TDMA as Medium Access Control Method Applied to ITS. Access Layer Part (2012)
14. ETSI TR 102 861 Intelligent Transport Systems (ITS); On the Recommended Parameter Settings for Using STDMA for Cooperative ITS. Access Layer Part (2012)
15. Scopigno, R., Cozzetti, H.A.: Mobile Slotted Aloha for Vanets. In: IEEE 70th Vehicular Technology Conference (2009)
16. Cozzetti, H.A., Scopigno, R.: Scalability and QoS in Slotted VANETs: Forced Slot Re-Use vs Pre-emption. In: The 14th International IEEE Conference on Intelligent Transportation Systems (2011)

17. Spaho, E., Barolli, L., Mino, G., Xhafa, F., Kolici, V.: VANET Simulators: A Survey on Mobility and Routing Protocols. In: The IEEE International Conference onBroadband and Wireless Computing, Communication and Applications, BWCCA (2011)
18. Weingartner, E., vom Lehn, H., Wehrle, K.: A performance comparison of recent network simulators. In: Proc. of the IEEE International Conference on Communications (2009)
19. Svilen, I., Andr, H., Georg, L.: Experimental validation of the ns-2 wireless model using simulation, emulation, and real network. In: 4th Workshop on Mobile Ad-Hoc Networks (2007)
20. NS-2 Network Simulator tool, http://www.isi.edu/nsnam/ns
21. NS-3 Network Simulator tool, http://www.nsnam.org/
22. Martinez, J.F., Toh, C.K., Cano, J.C., Calafate, C.T., Manzoni, P.: A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). The Journal Wireless Communications and Mobile Computing 11(7) (2011)
23. Pilosu, L., Fileppo, F., Scopigno, R.: RADII: A Computationally Affordable Method to Summarize Urban Ray-Tracing Data for VANETs. In: 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Wuhan, China, pp. 1–6 (September 2011)
24. Boban, M., Vinhoza, T.T.V., Ferreira, M., Barros, J., Tonguz, O.K.: Impact of Vehicles as Obstacles in Vehicular Ad Hoc Networks. The IEEE Journal on Selected Areas in Communications 29(1), 15–28 (2011)
25. ETSI TS 102 637-2. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service (2011)
26. Khajeh-Hosseini, A., Sommerville, I., Sriram, I.: Research Challenges for Enterprise Cloud Computing. CoRR Journal, abs/1001.3257 (2010)
27. Rymer, J.R., Gualtieri, M.: Cloud Computing Brings Demand for Elastic Application Platforms (2011), http://www.cloudsoftcorp.com/wp-content/uploads/FORRESTER-REPORT-Elastic-Application-Platforms.pdf
28. Amazon, Inc., Amazon Elastic Compute Cloud, User Guide, API Version 2011-12-15 (2012), http://awsdocs.s3.amazonaws.com/EC2/latest/ec2-ug.pdf
29. Chierici, A., Veraldi, R.: A quantitative comparison between xen and kvm. In: Proc. of 17th International Conference on Computing in High Energu and Nuclear Physics. Journal of Physics: Conference Series, vol. 219 (2010)
30. Amazon, Inc., Amazon Elastic Compute Cloud (Amazon EC2) (December 2008), http://aws.amazon.com/ec2/
31. Iosup, A., Ostermann, S., Yigitbasi, N., Prodan, R., Fahringer, T., Epema, D.: Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. Proc. of the IEEE Transactions on Parallel and Distributed Systems 22(6), 931–945 (2011)
32. Amazon, Inc., Amazon Elastic Block Store (EBS) (2012), http://aws.amazon.com/ebs/
33. Bilstrup, K., Uhlemann, E., Strom, E.G., Bilstrup, U.: On the Ability of the 802.11p MAC Method and STDMA to support Real-Time Vehicle-to-Vehicle Communication. EURASIP Journal on Wireless Communication and Networking 2009 (2009)

Constructing Large Scale Cooperative Multi-Agent Systems from Semantic P2P Networks

Lican Huang

Abstract. When we construct ”Smarter Planet” by gathering all information and cooperating all objects of physical and digital world in the era of Internet of Things, the agents as hubs and gateways play very important roles. As the Network becomes huge scale, large scale multi-agent systems become more and more important. Multi-agent systems play important roles in various applications. However, large scale multi-agent systems encounter many big issues such as scalability and inter-operability. This chapter presents a different approach of large scale multi-agent system architecture based on semantic P2P Network– Virtual Hierarchical Tree Grid Organizations (VIRGO) in mathematic terms. Other than unstructured and DHT-based structured P2P networks, VIRGO keeps the semantic meanings of the nodes according to their roles in the communities. In VIRGO approach, the agents are identified as domain names classified by the semantic meaning of roles in the organizations. The agents construct the VIRGO network by joining the virtual groups with the same names as their domains. We proof by mathematics that in this approach each agent has at least one path to communicate with any other agent and the performance is effective as maximin hops is less than log(N), where N is total number of agents. The performance bottle-neck of tree-like topology can be avoided by cache policies. Here, we also introduce a SQL-like language for inter-operation among agents. The application cases of large scale multi agent systems based on semantic P2P networks are also described.

1 Introduction

Recently, the term ”Internet of Things” [1] becomes very hot. We can construct ”intelligent earth” like the project ”Smarter Planet” [2] by gathering all information

Lican Huang
Institute of Networking & Distributed Computing
Zhejiang Sci-Tech University, Hangzhou, Zhejiang, China, 310018
e-mail: licanhuang@zstu.edu.cn
Hangzhou Domain Zones Technology Co. Ltd., Baiyang Jiedao, Xiasha, Hangzhou, Zhejiang, China, 310018
e-mail: lican.huang.hz@gmail.com

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 257–277.
DOI: 10.1007/978-3-642-34952-2_11

and co-operating all objects of physical and digital world in the future. All objects are uniquely identifiable by the IDs such as Radio-frequency identification (RFID) and URI[3], and can be virtually represented in the Network. Various information of physical world gathered by various sensors and information generated by computers or input by Human Beings are collected by hubs and gateways, which distribute and cooperate the information among all the objects in the Network. In the infrastructure of IOT, the functions of these hubs and gateways can be implemented by agents. Furthermore, based on the information and rules, the agents may produce action commands which change the status of the other objects. Therefore, agents play very important roles in the ”Internet of Things”. As the Network become huge scale, large scale multi-agent systems (MASs) become more and more important.

A multi-agent system is a system composed of multiple interacting intelligent agents. In MASs , a single agent can not solve the problem solely due to the incomplete information or capabilities, thus, multi agents must work cooperatively; As there is no system global control and data are distributed, thus, effective communication among agents is required. When MASs are small scale, it is easy to satisfy the above requirements. However, as the Internet becomes prerequisite of human life, large scale Multi-Agent Systems (MASs) on the Internet become more and more important. This Internet scale multi agent systems can be used in many applications such as gaming , social simulations, etc. But, this Internet scale MASs will encounter many big issues such as scalability and inter-operability.

The potential solution for these problems of Internet scale MASs is adoption of P2P technologies. The P2P technologies are classified into two kinds. The unstructural P2P technology such as Freenet[4] using flooding way has shortage of heavy traffic and un-guaranteed search. The structural P2P technology using DHT such as Chord [5] loses semantic meaning. Due to the lacks of nodes' semantic meanings, the cooperation for the large scale multi-agents encounters big problems. Unlike traditional P2P technologies , this chapter presents a different approach of large scale multi-agent systems based on semantic P2P Network – Virtual Hierarchical Tree Grid Organizations (VIRGO) [6][7][8][9][10] .Other than unstructured and DHT-based structured P2P networks, VIRGO keeps the semantic meanings of the nodes' roles in the communities. In VIRGO approach, the nodes are identified as domain names classified by the semantic meaning of roles in the organizations. The nodes construct the VIRGO network according to their domains, which form a coalition of vertical virtual organizations. In every node, on the top of VIRGO an intelligent agent is implemented to collect and analyze the related knowledge information from other agents and send action commands to other agents. Each agent can communicate with others for several intermediate hops. The agent use SQL-like language to inquiry or act by the complex conditions.

The rest of this chapter is as follows: section 2 describes Large Scale Multi Agent Systems Architecture based on Semantic P2P Network; section 3 presents Formal Definition of Large Scale Cooperative Multi-Agent Systems, section 4 describes application cases of large scale Cooperative multi agent system based on semantic P2P networks, and finally we give conclusions.

2 Large Scale Multi Agent System Architecture Based on Semantic P2P Network

Virtual Hierarchical Tree Grid Organizations (VIRGO) [6] is a domain-related hierarchical structure hybridizing un-structural P2P and structural P2P technology. VIRGO consists of prerequisite virtual group tree, and cached connections. Virtual group tree is similar to VDHA [11] , but with multiple gateway nodes in every group. Virtual group tree is virtually hierarchical, with one root-layer, several middle-layers, and many leaf virtual groups. Each group has N-tuple gateway nodes. In VIRGO network, random connections cached in a node's route table are maintained. These cached connections make VIRGO a distributed network, not just a virtual tree network like VDHA. With random cached connections, the net-like VIRGO avoids overload in root node in virtual tree topology, but keeps the advantage of effective message routing in tree-like network. As the change of contents in route table, VIRGO uses different lookup protocol and maintenance protocols from VDHA(as Figure 1 shown).

In the architecture of VIRGO for multi-agent systems[12], nodes, each of which implements one intelligent agent [13], construct multi-tuple virtual hierarchical tree topology (as Figure 2 shown).

2.1 Formal Definitions

Definition 1. Agent: Software hosted in computers, which can automatically activate its functions according to environment, with Universal Unique Identification denoted as **AgentUUID**, and several **AgentVGID**s, which are hierarchical domain names(or sometimes called as nodeID). For example, licanhuang@zstu.edu.cn indicating that Dr. Lican Huang is working with Zhejiang Sci-Tech University in China, and licanhuang@bupt.edu.cn indicating that he is also a guest researcher of Beijing University of Posts and Communications.

Agent set A = { a_i , i = 1...n },
where a_i is an agent.

Definition 2. Virtual Group: Virtual group is formed virtually by agents according to their interests or roles in the communities.

All Virtual Group set (VG)
$VG = \{g_{ij}, i = 1...n, j = 1...m\}$,
where i is layer of virtual group, and j is the order of virtual group in the layer.
Virtual Group Subset (VGS)
$VGS \subseteq VG$
groupID: every group has ID which is sub domain name.
$VGID = \{gID_{ij}, i = 1...n, j = 1...m\}$,
That is, g_{ij} has groupID gID_{ij}. Here, VGID is the set of all groupIDs.

For example, cn is root virtual group ID. domains such as edu.cn, zstu.edu.cn are IDs of the offspring groups.

parent group , son group, offspring groups

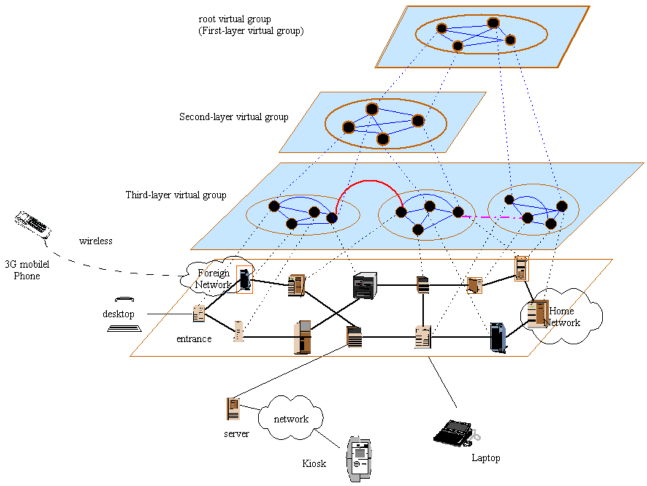


Fig. 1 Topology of Virtual Hierarchical Tree Grid Organizations

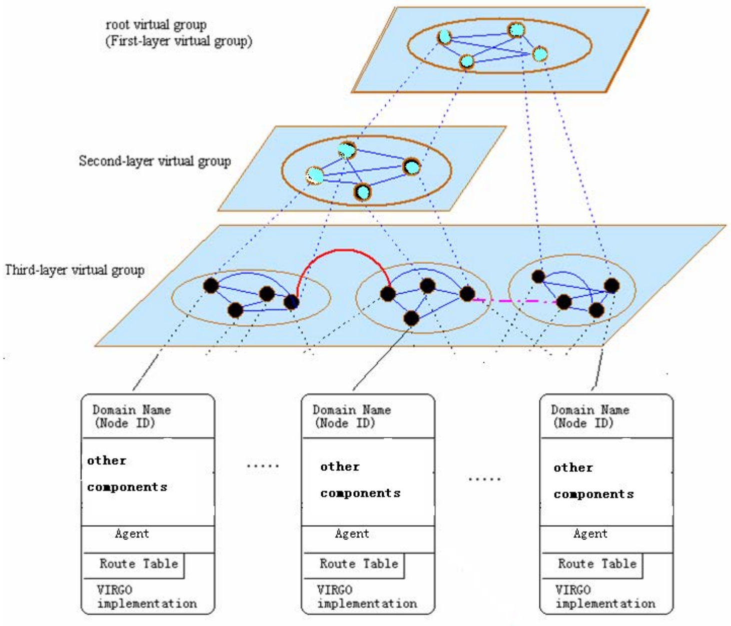


Fig. 2 Topology of large scale multi agents based on Semantic P2P Networks

When groupID gID_{ij} is the maximum substring of $gID_{i+1,l}$, $g_{i+1,l}$ is called as son groups of g_{ij}; $g_{ij}$is called as parent of $g_{i+1,l}$. Offspring groups are those groups whose substring of gIDs contain gID_{ij},

g_{11} is root virtual group without parent group, denoted as RVG. In the above example, its groupID gID_{11} is cn. It has son g_{21} whose groupID gID_{21} is edu.cn. g_{21} has son g_{31} whose groupID gID_{31} is zstu.edu.cn. g_{21} has a parent group g_{11}. edu.cn and zstu.edu.cn are son groups of cn. When g_{ij} has no offspring groups, it is called as leaf group, denoted as lg_{ij}. In the above example, the leaf groupID is zstu.edu.cn

Definition 3. offspring: the function to find the subset of offspring groups for a group.
$offspring(g_{ij}) = \{g_{i+l,k} | subString(gID_{i+l,k}) \supset gID_{ij}, k = 1...m, l = 1...n-i\}.$

Definition 4. son: the function to find the subset of son groups for a group.
$son(g_{ij}) = \{g_{i+l,k} | isMinLength(gID_{i+l,k}, g_{i+l,k} \in offspring(g_{ij}))\}.$

Definition 5. isWithinGroup: the function to judge the agent a with the state of having joined the virtual group g_{ij}.
$isWithinGroup(a, g_{ij}) = \{true, false\}$
if $isWithinGroup(a, g_{ij})$ == true , denoted by $a \multimap g_{ij}$;
if $isWithinGroup(a, g_{ij})$ == false , denoted by $a \nmid g_{ij}$;

Definition 6. isCached: the function to judge agent b cached by agent a into agent a's route table.
$isCached(a, b) = \{true, false\}$
if $isCached(a, b)$ = true , denoted by $a \curvearrowright b$;

Definition 7. Subgroup: the function to find a son group of g_{ij} which agent a joins group g_{ij} and this son group in the same time.
$subgroup(g_{ij}, a) = son(g_{ij}) | (a \multimap son(g_{ij})) \bigcap (a \multimap g_{ij});$

Definition 8. upgroup: the function to find the parent group for a group
$upgroup(g_{ij}) = g_{i-1,k} | i > 1, g_{ij} \in son(g_{i-1,k});$

Definition 9. agent: the function to find the set of agents joining g_{ij}
$agent(g_{ij}) = \{a_k | (a_k \multimap g_{ij}), k = 1...q\}$

Definition 10. Directed Connection: Agent a can communicate with agent b directly. That is, agent a knows agent b address.
isDirectConnect: the function to judge if agent a communicates agent b directly.
$isDirectConnect(a, b) = \{true, false\}$
if $isDirectConnect(a, b)$ == true , denoted by $a \longrightarrow b$;

Definition 11. cacheconnect: the function to find the set of agents directly connected by agent a due to cached information such as agent a's IP address.
$cacheconnect(a) = \{a_i | (a \curvearrowright a_i), i = 1...k\}$

Definition 12. treeconnect: the function to find the set of agents directly connected by agent a for tree topology (N-tuple virtual group tree).
$treeconnect(a) = \{\sum(agent(g_{ij}) | a \in agent(g_{ij}), i = 1...n, j = 1...m\}$

Definition 13. routetable: the function to find the set of agents listed in the agent a's route table. An agent keeps all directed connections with other agents in its route table.

$routetable(a) = \{treeconnect(a) \bigcup cacheconnect(a)\}$

Definition 14. Gateway Agent: gateway agent joins several different layers of virtual groups.

$gatewayagenta = a \in agent(g_{ij}) \bigcap a \in agent(upgroup(g_{ij})) \bigcap \ldots$

Definition 15. N-tuple Virtual Group Tree: N-tuple virtual group tree is a hierarchical tree formed by virtual groups. Among the agents of the lower layer virtual groups, N-tuple gateway agents in each group are chosen to form upper-layer groups, and from the agents of these upper- layer groups to form upper-upper-layer groups in the same way, and this way is repeated until a root-layer group is formed.

N-tuple Virtual Group Tree: satisfies the following conditions:

1. VG set is tree T,

(a) g_{11} is root RVG;

(b) $\exists upgroup(g_{ij}) \bigcap (|upgroup(g_{ij})| = 1)$, where $g_{ij} \in VG, i = 2 \ldots n, j = 1 \ldots m$;

2. $\forall((a_k \multimap g_{ij}) \bigcap (a \multimap g_{ij}))a \longrightarrow a_k$, $k = 1 \ldots |agent(g_{ij})|, i = 1 \ldots n, j = 1 \ldots m$;

3. if $|agent(g_{ij})| >= N_{tuple}$ then

$|\{a_k\}| = N_{tuple}$, where $(a_k \multimap upgroup(g_{ij})) \bigcap (a_k \multimap g_{ij}), i = 2 \ldots n, j = 1 \ldots m, k = 1 \ldots l$

else

$|\{a_k\}| = |agent(g_{ij})|$, where $(a_k \multimap upgroup(g_{ij})) \bigcap (a_k \multimap g_{ij}), i = 2 \ldots n, j = 1 \ldots m, k = 1 \ldots l$

Definition 16. VIRGO Topology: VIRGO topology is 4-tuple(VG, A, N_{tuple}, Cache), where VG is set of virtual groups, A is set of agents, N_{tuple} is n tuple replicated virtual tree, and Cache is the set of cached nodes.

VIRGO topology satisfies the following conditions:

1. N-tuple virtual group tree;
2. definitely, $treeconnect(a) \subseteq routetable(a), \forall a \in A$;
3. optionally , $cacheconnect(a) \subset routetable(a), \forall a \in A$.

Definition 17. Path

Let a, b_i and c are agents, where i=1...k. if $a \longrightarrow b_1$ or $a \curvearrowright b_1$, $b_1 \longrightarrow b_2$ or $b_1 \curvearrowright b_2$, ... $b_{k-1} \longrightarrow b_k$ or $b_{k-1} \curvearrowright b_k$, and $b_k \longrightarrow c$ or $b_k \curvearrowright c$, then there exists a path a to c , denoted by $a \rightarrowtail c$. Note: b_i may be optional.

Definition 18. Direct Path

Let a, b_i and c are agents, where i=1...k. if $a \longrightarrow b_1$, $b_1 \longrightarrow b_2$, ... $b_{k-1} \longrightarrow b_k$, and $b_k \longrightarrow c$, then there exists a direct path a to c, denoted by $a \longmapsto c$. Note: b_i may be optional.

2.2 Theorems Related to VIRGO

Lemma 1. (**Path Transitive**)
If $a \rightarrowtail b$ and $b \rightarrowtail c$, then $a \rightarrowtail c$.

Proof. Let a, $v_i, i = 1...k$, $u_i, i = 1...l$ and c are agents. Supposing there are intermediary $v_i, i = 1...k$ in $a \rightarrowtail b$ and intermediary $u_i, i = 1...l$ in $b \rightarrowtail c$. Due to $a \rightarrowtail b$, $a \longrightarrow v_1$ or $a \curvearrowright v_1$, $v_1 \longrightarrow v_2$ or $v_1 \curvearrowright v_2$, ... $v_{k-1} \longrightarrow v_k$ or $v_{k-1} \curvearrowright v_k$, and $v_k \longrightarrow b$ or $v_k \curvearrowright b$. In the same way, Due to $b \rightarrowtail c$, $b \longrightarrow u_1$ or $b \curvearrowright u_1$, $u_1 \longrightarrow u_2$ or $u_1 \curvearrowright u_2$, ... $u_{l-1} \longrightarrow u_l$ or $u_{l-1} \curvearrowright u_l$, and $u_l \longrightarrow c$ or $u_l \curvearrowright c$. By the definition 17, therefore $a \rightarrowtail c$.

Lemma 2. (**Bidirectional Direct Connection**)
In N-tuple virtual group tree, if $a \longrightarrow b$, then $b \longrightarrow a$.

Proof. Let $a \multimap lg_{kl}$ and $b \multimap lg_{mn}$. If $lg_{kl} = lg_{mn}$, then $b \longrightarrow a$ by definition the condition 2 of the definition 15. If $lg_{kl} <> lg_{mn}$, $a \longrightarrow b$ means $\exists g_{op} | a \in g_{op} \bigcap b \in g_{op}$. Therefore, $b \longrightarrow a$. Note: g_{op} is upgroup of both lg_{kl} and lg_{mn}.

Lemma 3. (**Bidirectional Direct Path**)
In N-tuple virtual group tree, if $a \longmapsto c$, then $c \longmapsto a$.

Proof. Let a, $b_i, i = 1...k$ and c are agents. Supposing there are intermediary $b_i, i = 1...k$ in $a \longmapsto c$. Due to $a \longmapsto c$, $a \longrightarrow b_1$, $b_1 \longrightarrow b_2, ...b_{k-1} \longrightarrow b_k$, and $b_k \longrightarrow c$. By the lemma 2, $c \longrightarrow b_k$, $b_k \longrightarrow b_{k-1}$, ..., $b_2 \longrightarrow b_1$, and $b_1 \longrightarrow a$. Therefore, $c \longmapsto a$

Lemma 4. (**Path to Root**)
In N-tuple virtual group tree, $\forall a_i, \exists r \in agent(RVG), a_i \longmapsto r$.

Proof. Let $a_i \in agent(lg_{kl})$, $\exists b | b \multimap lg_{kl} \bigcap b \multimap upgroup(lg_{kl})$ by the condition 3 of the definition 15. (Here, b may be a when $|agent(lg_{kl})| = 1$). Thus, $a_i \longrightarrow b$. $\exists c | c \multimap upgroup(lg_{kl}) \bigcap c \multimap upgroup(upgroup(lg_{kl})$ for the same reason, thus, $b \longrightarrow c$. After finite steps in the same way, we can reach RVG. That is: $\exists r \multimap RVG$, $u \longrightarrow r$. So, $a_i \longrightarrow b$, $b \longrightarrow c$, ..., $u \longrightarrow r$. By direct path definition 18, $a_i \longmapsto r$.

Theorem 1. (**Connection's Theorem**)
Let set A formed into **VIRGO topology** *defined by the definition 16, then every two agents (a,b) exist at least one path. $a \longmapsto b$ and $b \longmapsto a$.*

Proof. By lemma 4, $\exists r \in agent(RVG)$, $a \longmapsto r$; and $\exists w \in agent(RVG)$, $b \longmapsto w$,by lemma 3,$w \longmapsto b$; Mention that r and w belong to the same set (agent(RVG)), by the condition 2 of the definition 15 , $r \longmapsto w$. Therefore, $a \longmapsto r$, $r \longmapsto w$, $w \longmapsto b$. Therefore, by lemma 1, $a \longmapsto b$. By lemma 3, $b \longmapsto a$.

Theorem 2. (**Random Trace's Theorem**)
Let set A formed into **VIRGO topology** *defined by definition 16, then every two agents (a,b) communicate as a random path trace.*

Proof. Let a, v_i and b are agents, where i=0...k. v_i are intermediary agents which are gateways of communications. Route table of agent a contains treeconect and ramdom cacheconnect agents. $\forall a_i \in routetable(a)$, $a_i \longmapsto b$ by Theorem 1 (Connection's Theorem). Thus, the different numbers of paths which can be chosen is about $|routetable(a)|$ at agent a. In the same way, the different numbers of paths which can be chosen is about $|routetable(v_i)|$ at agent v_i. Therefore, the total number of paths is:

$|routetable(a)| + \prod_{i=0}^{i=k} |routetable(v_i)|$.

Therefore, the communication path would be random.

Theorem 3. (**VIRGO Complexity 's Theorem**)
Let set A formed into **VIRGO topology** *defined by definition 16, then complexity is effective.*

Proof. 1. **Space Complexity**

space complexity = $|routetable(a_i)|$,where i=0...n.

By the definition 16,

$|routetable(a_i)| = |treeconect(a_i)| + |cacheconect(a_i)|$.

$|treeconect(a_i)| = \sum_{k=h}^{k=p} |agent(g_{kl})|$, where $a_i \in agent(g_{kl}), k = h..p$

So, maximum of $|treeconect(a_i)|$ is $tree\_depth\times$ maximum of $|agent(g_{ij})|$.

Because of Tree topology,

maximum of $|treeconect(a_i)| = tree\_depth \times \log_{|g_{ij}|}(|A|)$

2. **Time Complexity**

As the same of proof process for Connection's Theorem, the hops of path for every two agents(a,b)is less than $2 \times tree\_depth$.

The tree depths are about:

$tree\_depth = \log_{|g_{ij}|}(|A|)$

Therefore, VIRGO complexity is effective.

2.3 Construction and Maintenance of VIRGO_Agent

In VIRGO_agent, all Agents are formed into N-tuple virtual organization tree. The Virtual Hierarchical Overlay Network can be established by manual or automatically by establishment protocol.

Definition 19. operation **join:** Agent a joins the virtual group g_{ij}.

$a \Longrightarrow g_{ij}$

Definition 20. operation **createnewgroup:** Agent a create new virtual organization g_{ij} .

$a \triangleright g_{ij}$

Definition 21. function **groupset(agent a):** to find all possible groups from Agent a's AgentVGID .

groupset(a) = $\{g_{ij} | gID_{ij}$ is substring of a's AgentVGID $\}$

Definition 22. function **group(agent a):** to find all groups which agent a is within .
group(a) = $\{g_{ij}|a \multimap g_{ij}\}$

Definition 23. function **send(message mes, agent a):** send message to agent a.

Definition 24. function **sendgroup(message mes, agentset** $\{a_i\}$):
send message to agent group a_i, i =1...k.

Definition 25. function **shareprefixs (agent a, agent b):** find groups which share same prefixes between agent a and b's AgentVGID .
$shareprefixs(a,b) = \{g_{ij}|g_{ij} \in groupset(a) \bigcap g_{ij} \in groupset(b)\}$

Definition 26. function **minLenthgroup (VGS):** find groups which have minimum length of groupID among group subset VGS.
minLenthgroup(VGS) = $\{g_{ij}\}|isMinumLength(gID_{ij}) \bigcap g_{ij} \in VGS$

Definition 27. function **maxLenthgroup (VGS):** find groups which have maxium length of groupID among group subset VGS.
maxLenthgroup(VGS) = $\{g_{ij}\}|isMaxiumLength(gID_{ij}) \bigcap g_{ij} \in VGS$

Definition 28. function **maxshareprefixs (agent a, agent b):** find group which shares same maximum prefixes between agent a and b's AgentVGID .
$maxshareprefixs(a,b) = g_{ij}|g_{ij} \in groupset(a) \bigcap g_{ij} \in groupset(b) \bigcap g_{ij} \in maxLenthgroup(shareprefixs(a,b))$

Definition 29. function **lookupfirstagent:** to find the first agent which shares the same virtual group with joining agent.
lookupfirstagent(a_{join}) = $a \in agent(g_{ij})|g_{ij} \in maxshareprefixs($ a_{join},a);

Definition 30. function **difgroups:** to find the different group set between agents a and b.
difgroups(agent a, agent b) = $\{g_{ij}|g_{ij} \in groupset(a) \bigcap g_{ij} \notin groupset(b)\}$

Definition 31. function **lookagent:** to find out agent a which a is within group o: that is: $a \multimap o$.
lookagent(group o) = $a|a \in agent(o)$

Definition 32. function **checkIsGroup:** to find out whether agent a is within group o; that is: $a \multimap o$ or $a \nmid o$.
$checkIsGroup(agenta, groupo) = \{a \multimap o, a \nmid o\}$

2.3.1 Setup New VIRGO_Agent

The Setup New VIRGO_Agent Algorithm establishes a new VIRGO_Agent network.

Algorithm 1. Setup New VIRGO_Agent Algorithm

```
$\forall g_{ij} \in groupset(a)\ a \triangleright g_{ij}, a \Longrightarrow g_{ij};$
```

2.3.2 Agent Join Protocol

When a Agent (*a_join*) joins VIRGO_agent, it first connects with an entrance agent(ent), through entrance agent it finds one of Agents which shares the maximum prefixes with the joining Agent, then the joining Agent sends the JOINMESSAGE to the latter,the latter will broadcast the message to all Agents in the virtual organization which *a_join* will join. Agent Join Protocol Algorithm describes a new Agent *a_join* joins VIRGO_agent.

2.3.3 Agent Leave Protocol

In the following, the Leave protocol describes a Agent a_{leave} leaves the network.

Algorithm 2. Agent Join Protocol Algorithm

```
1. $a_{join}.send(JOINMESSAGE, ent);$
2. $a = ent.lookupfirstagent(a_{join});$
3. $a_{join}.send(JOINMESSAGE, a));$
4. $\forall o \in difgroups(a_{join}, a) a_{join} \triangleright o;\ a_{join} \Longrightarrow o);$
5. $o_{ij} = maxshareprefixs(a_{join}, a);$
6. do {
  $a.sendgroup(JOINMESSAGE, agent(o_{ij});$
  $\forall a_l \in agent(o_{ij}) \{$
    $a_l.send(CONFIRMATIONMESSAGE, a_{join});$
    $a_l.RouteTableAdd(a_{join});$
    $a_{join}.RouteTableAdd(a_l);\ \}$
}
7.   $if\ o_{ij} == RVO\ break;$
8. $prevg = o_{ij}; o_{ij} = up(o_{ij});$
9. $a = a_i \in agent(o_{ij});$
$\}\ while\ |a_k \in agent(o_{ij}) \bigcap a_k \in agent(prevg)| < n_{tuple};$
```

Algorithm 3. Agent Leave Protocol Algorithm

```
$leftO = o \in group(a_{leave})|isMinLength(oID);$
do {
$\forall a_i \in agent(leftO) \{$
  $a_{leave}.send(LEAVEMESSAGE, a_i);$
  $a_i.send(CONFIRMATIONMESSAGE, a_{leave});$
  $a_i.RouteTableDelete(a_{leave});$
  $set\ leftO = subgroup(leftO);$
) $while(!leafVirtualorganization(leftO));$
```

2.3.4 Fault Agent Discovery Protocol

The Fault Agent Discovery Protocol describes a fault Agent a_{fault} being discovered by other agent $a_{discovery}$.

2.3.5 Fault Agent Replacement Protocol

The Fault Replacement Protocol describes that a fault Agent a_{fault} discovered by a agent $a_{discovery}$ is replaced by other agent $a_{replacement}$.

Algorithm 4. Fault Agent Discovery Protocol-Heartbeat Algorithm

```
do {
 $\forall g_{ij}\{$
   $a = random(a_k \in agent(g_{ij}));$
   $a.sendgroup(ALIVEREQMESSAGE, agent(g_{ij}));$
   $\forall a_l \in agent(g_{ij})\{$
   $a_l.send(ALIVRESPEMESSAGE, a)\};$
   $do\{$
     $timesleep(nseconds); times++;$
     $a.sendgroup(ALIVEREQMESSAGE, noresponse(agent(g_{ij})));$
   $\}while(times < TryTimes);$
   $a_{discovery}$ = a;
   $a_{fault} = a_l \in noresponse(agent(g_{ij}))$ ;
}
}
```

Algorithm 5. Agent Fault Replacement Algorithm

```
$o = a_{discovery}.maxLenthgroup(group(a_{fault}));$
$b = a_{discovery};$
$do\{$
 $a = b.lookagent(o);$
 $b.send(RepalceMessage, a);$
 $o_l = o_l \in Subgroup(o) \bigcap o_l \in groupset(a_{fault});$
 $if(a.checkIsGroup(a_{fault}, o))\{$
   $a_l = a.lookagent(o_l) \bigcap a_l \nmid o;$
   $a_l \Longrightarrow o;$
   $a = a_l;$
 $\}$
 $b = a;$
 $o = upgroup(o);$
$\}\ while(o \neq RVG);$
```

3 Formal Definition of Large Scale Cooperative Multi-Agent Systems Based on Semantic P2P Networks

Agent has order action to ask other agents to execute some actions; execution action ordered by other agent ; inquiry action to ask other agents to answer information; answer action to answer other agent's inquiry. The formal definitions are given as the following:

As before, A is set of agents. VG is whole set of groups.

Definition 33. operation **order:** agent a orders agent b to execute(a and b can be the same agent), denoted by:
$a \twoheadrightarrow b$
$orderset(O) = \{a_i \twoheadrightarrow b_j, a_i \in A \bigcap b_i \in A\}$

Definition 34. operation **grouporder:** agent a orders agents belong to virtual group g_{ij} who satisfy search_condition to execute. denoted by:
$a \twoheadrightarrow agent(g_{ij})$
$grouporderset(GO) = \{a_i \twoheadrightarrow agent(g_{kl}), a_i \in A \bigcap g_{kl} \in VG\}$

Definition 35. operation **execution:** agent a execute action ordered by other agent. denoted by:
$a \twoheadleftarrow b$
$executionset(E) = \{a_i \twoheadleftarrow b_j, a_i \in A \bigcap b_i \in A\}$

Definition 36. operation **inquiry:** agent a inquiries information of agent b . denoted by:
$a \looparrowright b$
$inquiryset(I) = \{a_i \looparrowright b_j, a_i \in A \bigcap b_i \in A\}$

Definition 37. operation **groupinquiry:** agent a inquiries information of agents belong to virtual group g_{ij} who satisfy search_condition . denoted by:
$a \looparrowright agent(g_{ij})$
$groupinquiryset(GI) = \{a_i \looparrowright agent(g_{kl}), a_i \in A \bigcap g_{kl} \in VG\}$

Definition 38. operation **answer:** agent a answers information of agent b. denoted by:
$a \looparrowleft b$
$answerset(W) = \{a_i \looparrowleft b_j, a_i \in A \bigcap b_i \in A\}$

Definition 39. **ACT set** is the union of all action sets.
$ACT = \{O \bigcup GO \bigcup E \bigcup I \bigcup GI \bigcup W\}$

Definition 40. **State set S** defines all states of all agents. **State set** S_i defines all states of agents a_i. S = { S_i, i =1...n}

Definition 41. **VIRGO_agent** is large scale multiagent system which is defined as 8-tuples:
$VIRGO\_agent = \{A, VG, N_{tuple}, Cache, S, ACT, SFn, AFn\}$

Here, A,VG,N_{tuple}, and Cache which define VIRGO topology according to the definition 16.

$SFn := S \times ACT \longrightarrow S;$

$AFn := S \longrightarrow ACT;$

3.1 Distributed Action for Large Scale Cooperative Multi-Agent Systems

The agents in large scale multi-agent systems can inquiry other agents' information and tell other agents how to act. Sometimes, we do not know any information except action conditions. This leads to a big issues in traditional large scale multiagent systems.

We use semantic P2P networks and SQL-like act language to solve the problem. In VIRGO_agent system , the AgentVGID is defined as the local-part@global-hier-part. The global part is similar as domain name. The local part is local name such as IBM. Unlike domain name in DNS, the global-hier-part of AgentVGID has the semantic meanings of the topics the users are interested in. For example, Alice is a fan of Britney Spears, she can join virtual group of Britney.popular.music; in the meantime, she is a staff of mit.edu, therefore, her AgentVGID is also Alice@mit.edu. The peers join virgo network according to their domain names (global-hier-part).

Agents use statements to query from and/or act other agents. The agents can use various communication languages. We prefer the SQL-like Act language as the following:

```
act::=  ACT  actscripts
        ON domainref
        [WHERE search_condition];
```

Here, ACT is the verbal for action; actscripts is the defined actions; domainref is for Domain Name, search_condition is similar to SQL statement in Database. The SQL-like Act language has advantage for acting subsets of agents with the specific properties.

The distributed action execution process is as the following: source peer (agent a_{source}) sends ACT MESSAGE, which contains SQL-like act statement defined above, to nearer peer. The nearer peer will route to the even nearer peer by calculating theoretical hops. Repeat this process, until the nearest gateway peer has been found. Finally, the nearest gateway peer broadcasts the message to the member peers in the virtual group; and the member peers will check their properties database(PDB). If the nodes match the search conditions, these nodes will execute the actions defined in the ACT message. Finally, the nearest gateway peer collects these result information and responds to the source peer. The detailed algorithm is listed as Distributed Act Algorithm.

Definition 42. mintheoryhop: the function to find an agent in route table which has minimum hops to destination group.

$mintheoryhop(a,g) = \{b | b \in routetable(a) \bigcap b$ with minimum hops to destination group g $\}$

Definition 43. parsergroup(ACTMESSAGE.domainref): resolve the group in sql-like act statement in ACTMESSAGE.

Definition 44. samegroup(AgentVGID, g_{ij}): boolean function to judge if two group names are equal.

Definition 45. a.matchup(ACTMESSAGE.search_condition): agent a checkup its property database with search_condition in sql-like act statement in ACTMESSAGE.

Algorithm 6. Distributed Act Algorithm

```
1.$a_{source}.send(ACTMESSAGE, a \in mintheoryhop(a_{source},$
  $group(ACTMESSAGE.domainref));$
2.do {
  $a.send(ACTMESSAGE, b \in mintheoryhop(a,$
    $parsergroup(ACTMESSAGE.domainref));$
  $set\ a = b;$
$\}while!samegroup(a.AgentVGID, parsergroup(ACTMESSAGE.domainref));$
3. $a.sendgroup(ACTMESSAGE,$
  $c_i \in agent(parsergroup(ACTMESSAGE.domainref));$
4. $\forall c_i \in agent(parsergroup(ACTMESSAGE.domainref))\{$
  $if c_i.matchup(ACTMESSAGE.search\_condition)\{$
  $c_i.execute(ACT);$
  $c_i.send(ResultMessage, a_{source});$
  }
}
```

4 Case Applications

4.1 Simulation of Social Society Based on Semantic P2P Network

Social sciences study aspects of human society. Traditionally, social sciences may use empirical methods or critical analysis to study social phenomenon. Some social scientists use critical theory to deduce knowledge by the examination and critique of society and culture such as dialectical materialism of Karl Marx. Other social scientists try to transform quantitative and qualitative aspects of social phenomena into numerical variables. They often use empirical techniques such as questionnaires, field-based data collection, archival database information and laboratory-based data collections, and use mathematical models to analyze the collected data or to simulate the social phonomania[14]. IT technologies are widely used in social science; one of those is computational social science. There are many analysis models to study society, however, today's society is too complex to analyze. Therefore, simulating social society is alternative for society study.

Here, we describe the large scale multi-agent systems based on semantic P2P networks–virtual hierarchical tree Grid organizations(VIRGO) to simulate social society with millions of computers and sensors. The virtual social persons which simulate human's actions and responses by computer algorithms, join virtual organizations as similar as real organizations. The nodes hosting virtual social persons construct n-tuple overlay virtual hierarchical overlay network(VIRGO) according to the domains of their joined virtual organizations.

For easy implementation, we use quantized virtual society to model the real society . The quantized virtual society is the virtual society in which things are quantized and the actions to be taken are chosen by scores among the candidate actions.

4.1.1 Definition of Quantized Virtual Society

Virtual society, which simulates real society, consists of virtual social person and organizations [9]. The quantized virtual society is the virtual society in which things are quantized and the actions to be taken are chosen by scores among the candidate actions. We give several definitions in the following:

Definition 46. SocialElementSET: the union of people set(personSET), organization set(organizationSET) and government set(governmentSET).

Definition 47. resourceSET: all useful resources people, organizations, governments held such as house, clothes, food, electricity, water, land, etc.

Definition 48. resourceTypeSET: The resources can be classified as private, public. Private resources can be classified as person owned, organization owned and government owned. The private resources can only be used by the owners, but it can be exchanged between owners. The public resources can be used by different people, organizations and governments. The resources can be also classified as mutual, which can only be used by one member of SocialElementSET at the same time, and sharable, which can be used by different members of SocialElementSET at the same time. The resources can be further classified as consumable, whose amount are reduced after use, unchangable, whose amount are unchanged after use , and earnable, whose amount are increased after use.

Let us define:
OwnTypeSet = {private,public}
useTypeSet = {mutual, sharable }
consumeTypeSet {consumable,unchangable,earnable}
Therefore,
resourceTypeSET = OwnTypeSet × useTypeSet × consumeTypeSet
Every resource has one element of resourceTypeSET as its attribute.

Definition 49. Wealth_value: resources quantized as wealth like money in real society.

quantizeWealthFun: the function to quantize the resources to wealth_value.
quantizeWealthFun: (resourceSET)$\longrightarrow R$;
R is real set.

Definition 50. power_value: power refers to the ability to influence or control others in organizations or governments. The power can be enlarged through hierarchical organizations.

quantizePowerFun: the function to quantize the resources and roles in organizations into power_value.

quantizePowerFun:(resourceSET,roleSET)$\longrightarrow R$;

R is real set.

Definition 51. role: the role or position in organization.

roleSET: all roles of all people in all organizations.

Definition 52. Quantized Virtual Social Person(QVSP): which simulates real person in Quantized society.

$QVSP = \{resourceHeld, need, desire, curRoles,$
$ability, wealth\_value, power\_value\}$

Here, resourceHeld is subset of resourceSET the person held or shared. The need is the subset of resourceSET plus minimal wealth_value and power_value the person needs to survive. The person's desire is subset of resourceSET plus wealth_value and power_value the person hope to obtain. The curRoles refers to the position in virtual social organizations the person held. Every role has attribute of life_value, which must be larger than 0. The ability is the set of functions referring to abilities to earn money, power,resources, etc. That is:

ability_func:=(resourceSET,roleSET,wealth_value,power_value)$\longrightarrow$
(resourceSET,roleSET,wealth_value,power_value);

Definition 53. SocialAgent: Software hosted in computers, which can automatically simulate person behaves by activating its functions according to environment, with Universal Unique Identification denoted as **AgentUUID**, and several **AgentVGID**s, which are hierarchical domain name(or sometimes called as nodeID). For example, alice@IBM.com.

SocialAgent set A = $\{a_i, i = 1...n\}$,

where a_i is an SocialAgent.

Definition 54. Quantized Virtual Social Person Set: all persons in the society, denoted by **QVSPSET**.

Definition 55. Quantized Virtual Social Organization(QVSO): which simulates real organizations in Quantized society.

$QVSO = \{goal, members, social\_relation,$
$architecture, resource, life\_value, wealth\_value,$
$power\_value\}$

Here, goal is the set of goals of the organizations. Members are subset of QVSPSET who join the organization. Social relation is the relations of virtual social persons. Architecture is organizations' architecture such as hierarchical tree structure. Resources refer to the resources held by the organizations. The virtual social organization has the life value which must be larger than 0, the wealth value which the organizations uses to exchange resources or power such as TAX, and the power value which the organizations use to affect virtual social persons within the organizations or other organizations.

Virtual society is an organization with sub organizations. The organizations may have hierarchical structures or other type structures. Virtual society is formed by virtual social organizations. The quantized virtual society is the virtual society in which things are quantized and the actions to be taken are chosen by scores among the candidate actions. So,

Definition 56. Quantized Virtual Society: which simulates real society.
$Quantized\_virtual\_society = \{status, score, QVSOSET, organization\_network\}$
Here, status is the salability of the society; the organization networks are networks formed by organizations. Score is measurement to judge the virtual social society. QVSOSET is set of QVSO.

We model quantized virtual social persons as agents[15]. They join virtual organizations of VIRGO equal to the quantized virtual social organizations.

Definition 57. VIRGO Virtual Society Topology: VIRGO virtual society topology is 4-tuple(VO, A, N_{tuple}, Cache), where VO is set of virtual organizations, A is set of SocialAgents, N_{tuple} is n tuple replicated virtual tree, and Cache is set of cached nodes.

VIRGO virtual society topology satisfies the following conditions:

1. N-tuple virtual organization tree;
2. definitely, $\forall$ a $\in$ A $\exists$ treeconnect(a) $\in$ routetable(a) ;
3. optionally , $\forall$ a $\in$ A $\exists$ cacheconnect(a) $\in$ routetable(a) .

4.2 Semantic P2P Network for e-Business

Traditionally, e-Business uses Client/Server technologies, in which the consumers and producers do businesses through the third parties like Alibaba.com. When most of the enterprises deploy management systems and Web Servers, the more wide scope of e-Business such as e-Business corporations of many partners for the whole supply chain is target of e-Business. However, the complex business activities are not easily modeled and implemented by several third parties. When a lot of Small to Medium Enterprises (SME) have their own Internet business servers and/or Enterprise Resource Planning systems, large scale multi-agent systems based on semantic P2P Network are the potential for the solution of the e-Business corporations. The large scale multi-agent systems based on semantic P2P Network can do distributed complex action execution by SQL-like statements.

Figure 3 shows the architecture of VIRGO for e-Business, in which two-tuple virtual hierarchical tree topology (the nodes in different layers connected with dash line are actually the same node). Every enterprise's node has an intelligent agent [13] which contains the component of VIRGO implementation. The agent will communicate with the other enterpriser management systems such as ERP,SCM, etc.

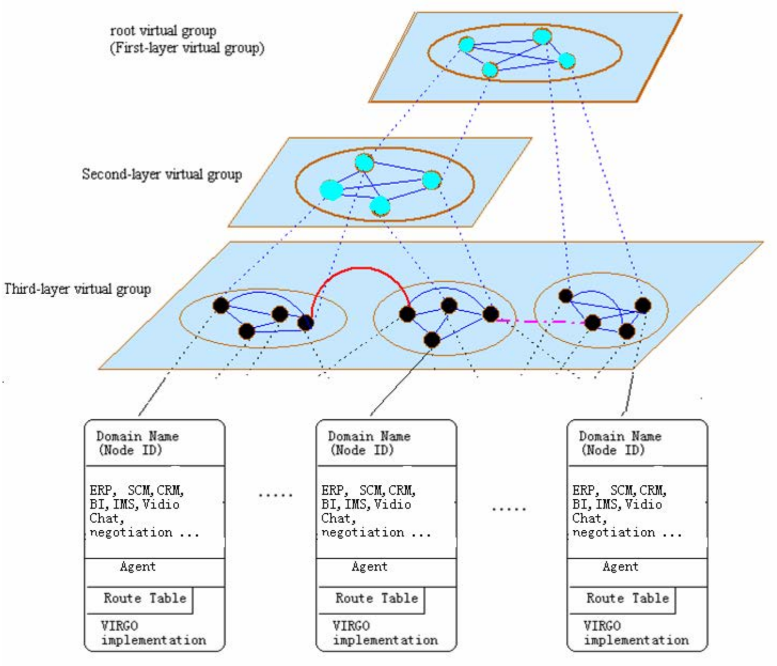

root virtual group
(First-layer virtual group)
Second-layer virtual group
Third-layer virtual group
Domain Name
(Node ID)
ERP, SCM, CRM, BI, IMS, Vidio Chat, negotiation ...
Agent
Route Table
VIRGO implementation
.....
Domain Name
(Node ID)
ERP, SCM, CRM, BI, IMS, Vidio Chat, negotiation ...
Agent
Route Table
VIRGO implementation
.....
Domain Name
(Node ID)
ERP, SCM, CRM, BI, IMS, Vidio Chat, negotiation ...
Agent
Route Table
VIRGO implementation

Fig. 3 Architecture of VIRGO for e-Business

All enterprises are classified according to a given enterpriser classification method such as International Standard Industrial Classification of All Economic Activities (Isic), Rev.4 [16].

Figure 4 shows the topology of some enterprises classified as ISICV4.

By using the SQL-like query language, the enterprise agents can finish tasks co-operatively.

Operation 1

To advocate information about new HAV vaccine products to all drug wholesaler, we can use the following Statement:

Advocate noticeContext On enterprise.wholesale.drugwholesale where prod =”HAVvaccine”;

Operation 2

To bid eggs with lowest price from all egg companies. We can use the following Statement:

Bid lowestPrice On enterprise. Agriculture-forestry-fishing.animalproduct where prod =”egg” and schedule = ”2009-08-20” and quality = ”best” and amount =”10000”;

Operation 3

To procure paper packages from all paper package factories, we can use the following Statement:

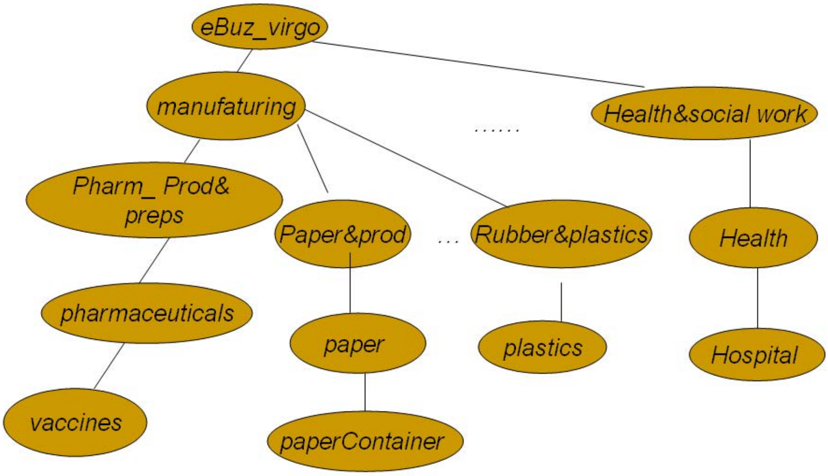


Fig. 4 Enterprises classifications

Procure Procurement-workflow-scripts1 On enterprise. Manufacturing.paper-and-paperproducts.paperContainer where prod =”HAVvaccinePackage” ;

5 Conclusion

This paper presents large scale multi-agent system(MAS) VIRGO_Agent based on semantic P2P Network– Virtual Hierarchical Tree Grid Organizations (VIRGO). The agents are identified as domain names classified by the semantic meaning of roles in the organizations , and construct the VIRGO network by joining the virtual groups with the same names as their domains. The performance of this approach is effective as maximin hops less than log(N), where N is total number of agents, and avoids the performance bottle-neck of tree-like topology by cache policies. We here give mathematical definitions for VIRGO_Agent, and proof by mathematics that in VIRGO_Agent each agent has at least one path to communicate with any other agent. We also define distributed Action for large scale multi-agent systems by introduced SQL-like language. Two case applications of large scale cooperative multi-agent system(MAS) VIRGO_Agent based on semantic P2P Networks are discussed.

Acknowledgements. This paper is supported by Hangzhou Domain Zones Information Technology Co.Ltd. The copyright of the software is owned by Hangzhou Domain Zones Information Technology Co.Ltd.

Term Index

Internet of Things(IOT): All things both in physical and digital world are tagged by unique identification Ids and can be represented virtually in the Network.
Radio-frequency identification (RFID): The unique tags of objects by using radio-frequency electromagnetic fields for the purposes of automatic identification and tracking.
Sensor: equipment gathers information of physical world.
Agent: Software hosted in computers or intelligent equipment, which can take roles of hub and gateway and automatically activate its functions according to environment.
Multi-agent system (MAS): system composed of multiple interacting agents within an environment.
Large scale multi-agent system : system composed of huge amounts of multiple interacting agents in the Network.
Virtual Hierarchical Tree Grid Organizations (VIRGO): a domain-related hierarchical structure by hybridizing un-structural P2P and structural P2P technology. Among the agents of the lower layer virtual groups, N-tuple gateway agents in each group are chosen to form upper-layer groups, and from the agents of these upper-layer groups to form upper-upper-layer groups in the same way, and this way is repeated until a root-layer group is formed.
Virtual group: Virtual group is formed virtually by agents according to their interests or roles in the communities.
Semantic P2P Network: a new type of P2P networks in which nodes are classified as DNS-like domain names with semantic meanings such as Alice @Brittney.popular.music. Semantic P2P networks contains prerequisite virtual tree topology and net-like topology formed by cached nodes. Semantic P2P networks keep the semantic meanings of nodes and their contents. The nodes within semantic P2P networks can communicate each other by various languages. Semantic P2P network can execute complicated queries by SQL-like language.
Large scale multi-agent system based on semantic P2P Network: system composed of huge amounts of multiple interacting agents whose IDs are represented by domain-like names. These agents form semantic P2P network according to their interests.
Distributed Action for Large Scale Cooperative Multi-agent System: the agents in large scale multi-agent systems can inquiry other agents' information and tell other agents how to act. Agents use statements to query from and/or act other agents. The agents can use various communication languages such as SQL-like Act language.
SQL-like Act language: The format of SQL-like Act language is as the following: act::= ACT actscripts ON domainref [WHERE search_condition]. Here, ACT is the verbal for action; actscripts is the defined actions; domainref is for Domain Name, search_condition is similar to SQL statement in Database.

References

1. Ashton, K.: That 'Internet of Things' Thing. RFID Journal 22 (July 2009)
2. http://www.ibm.com/smarterplanet/
3. http://tools.ietf.org/html/rfc3986
4. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: Federrath, H. (ed.) Anonymity 2000. LNCS, vol. 2009, pp. 46–66. Springer, Heidelberg (2001)
5. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications. In: The 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 149–160. ACM Press, New York (2001)
6. Huang, L.: VIRGO: Virtual Hierarchical Overlay Network for Scalable Grid Computing. In: Sloot, P.M.A., Hoekstra, A.G., Priol, T., Reinefeld, A., Bubak, M. (eds.) EGC 2005. LNCS, vol. 3470, pp. 911–921. Springer, Heidelberg (2005)
7. Huang, L.: A P2P service discovery strategy based on content catalogues. Data Science Journal 6, S492–S499 (2007)
8. Huang, L.: Resource Discovery Based on VIRGO P2P Distributed DNS Framework. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part III. LNCS, vol. 5103, pp. 501–509. Springer, Heidelberg (2008)
9. Huang, L.: A P2P Framework of Virtual Society based on Virtual Hierarchical Tree Grid Organizations(VIRGO). In: SP2PN 2009 in conjunction with NISS 2009, Beijing, China, pp. 1393–1396. IEEE CPS Publishing (2009), doi:10.1109/NISS.2009.189
10. Huang, L.: Locating Interested Subsets of Peers for P2PSIP. In: SP2PN 2009 in conjunction with NISS 2009, Beijing, China, pp. 1408–1413. IEEE CPS Publishing (2009), doi:10.1109/NISS.2009.85
11. Huang, L., Wu, Z., Pan, Y.: Virtual and Dynamic Hierarchical Architecture for e-Science Grid. International Journal of High Performance Computing Applications 17(3), 329–347 (2003)
12. Huang, L.: Large Scale Cooperative Multiagent System Based on Semantic P2P Network. In: ICNDC 2010 Proceedings of the 2010 First International Conference on Networking and Distributed Computing, Hangzhou, China, pp. 381–386. IEEE CPS Publishing (2010), doi:10.1109/ICNDC.2010.79
13. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn., ch. 2. Prentice Hall, Upper Saddle River (2003) ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/
14. http://en.wikipedia.org/wiki/Social_sciences
15. Jennings, N.R.: Agent-Based Computing: Promise and Perils. In: IJCAI 1999, pp. 1429–1436 (1999)
16. http://unstats.un.org/unsd/cr/registry/regcst.asp?Cl=27

An Efficient, Secure and User Privacy-Preserving Search Protocol for Peer-to-Peer Networks

Jaydip Sen

Abstract. A peer-to-peer (P2P) network is a distributed system in which the autonomous peers can leave and join the network at their will and share their resources to perform some functions in a distributed manner. In an unstructured P2P network, there is no centralized administrative entity that controls the operations of the peers, and the resources (i.e., the files) that the peers share are not related to the their topological positions in the network. With the advent of the Internet of Things (IoT), the P2P networks have found increased interest in the research community since the search protocols for these networks can be gainfully utilized in the resource discovery process for the IoT applications. However, there are several challenges in designing an efficient search protocol for the unstructured P2P networks since these networks suffer from problems such as fake content distribution, free riding, whitewashing, poor search scalability, lack of a robust trust model and the absence a of user privacy protection mechanism. Moreover, the peers can join and leave the network frequently, which makes trust management and searching in these networks quite a challenging task. In this chapter, a secure and efficient searching protocol for unstructured P2P networks is proposed that utilizes topology adaptation by constructing an overlay of trusted peers and increases the search efficiency by intelligently exploiting the formation of semantic community structures among the trustworthy peers. It also guarantees that the privacy of the users and data in the network is protected. Extensive simulation results are presented and the performance of the protocol is also compared with those of some of the existing protocols to demonstrate its advantages.

Keywords: P2P network, topology adaptation, trust, reputation, semantic community, malicious peer, user privacy.

Jaydip Sen
Innovation Labs, Tata Consultancy Services Ltd.
Bengal Intelligent Park, Salt Lake Electronic Complex, Kolkata 700091, India
jaydip.sen@acm.org

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 279–320.
DOI: 10.1007/978-3-642-34952-2_12

1 Introduction

During the past few years, in the area of wireless communications and networking, a novel paradigm named the IoT which was first introduced by Kevin Ashton in the year 1998, has gained increasingly more attention in the academia and industry [45]. By embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves, IoT would add a new dimension to the world of information and communication. Unquestionably, the main strength of the IoT vision is the high impact it will have on several aspects of every-day life and behavior of potential users. From the point of view of a private user, the most obvious effects of the IoT will be visible in both working and domestic fields. In this context, assisted living, smart homes and offices, e-health, enhanced learning are only a few examples of possible application scenarios in which the new paradigm will play a leading role in the near future [5][7]. Similarly, from the perspective of the business users, the most apparent consequences will be equally visible in fields such as automation and industrial manufacturing, logistics, business process management, intelligent transportation of people and goods.

The real power of the IoT lies in the universal connectivity among all devices and objects. However, it calls for interoperability so that the service requestors must know the features offered by the service providers, and it should be possible for the service requestors to understand what the service providers have to offer by semantic modeling. This is a key issue for stepping towards ubiquitous services, where the new or modified services may appear at any time, and towards device networks that are capable of dynamically adapting to the context changes as may be imposed by the application. This calls for a middleware which will interface between the devices and the applications. Since the devices need to communicate with each other, there is a need for a naming and addressing scheme, and a mechanism for search and discovery. Moreover, since each device is mapped to an identity (through naming and addressing), there are serious security and privacy concerns. All these challenges should be tackled by the middleware. One efficient approach for developing the middleware platform for IoT is using a multi-agent system. In a massively distributed system like the IoT, several agent platforms will exist each having a set of agents running and registered with a *directory facilitator* (DF). Problem, however, arises when the agents from different platforms will have to search the remote DFs and interact with the agents located on the remote platforms. In these scenarios, the agents will have to use resource discovery protocols which are similar to the file searching protocols in a purely unstructured P2P network. Hence, efficient searching in unstructured peer-to-peer network has a direct contextual relevance to the resource discovery in IoT applications. We provide below a brief descussion on the P2P networks before presenting the motivation and contribution of this chapter.

The term *P2P systems* encompasses a broad set of distributed applications which allow sharing of computer resources by direct exchange between systems. The goal of a P2P system is to aggregate resources available at the edge of Internet and to share it co-operatively among the users. The file sharing P2P systems have particularly become popular as a new paradigm for information exchange among large number of users in the Internet. These systems are more robust, scalable, fault-tolerant and they offer better availability of resources than the traditional systems based on the client-server model. Depending on the presence of a central server, the P2P systems can be classified as *centralized* or *decentralized* [44]. In the decentralized architecture, both the resource discovery and the resource download happen in a distributed manner. The decentralized P2P architectures may further be classified as *structured* or *unstructured* networks. In structured networks, there are certain restrictions on the placement of the contents and the network topologies. In unstructured P2P networks, however, the placement of the contents is unrelated to the topologies of the networks. The unstructured P2P networks perform better than their structured counterparts in dynamic environments. However, they need efficient search mechanisms and they also suffer from numerous problems such as: possibilities of fake content distribution, free riding (peers who do not share, but consume resources), whitewashing (peers who leave and rejoin the system in order to avoid penalties) and the lack of scalability in searching. The open and anonymous nature of the P2P applications leads to a complete lack of accountability of the contents that a peer may put in the network. The malicious peers often use these networks to carry out content poisoning and to distribute harmful programs such as Trojan Horses and viruses [47]. *Distributed reputation based trust management systems* have been proposed by the researchers to provide protection against the malicious content distribution in a distributed environment [1]. The main drawbacks of these schemes are their high overheads of message exchange and their susceptibility to misrepresentation by the malicious nodes. Guo et al. have proposed *trust-aware adaptive P2P topology* to control the free-riders and the malicious peers [29]. In [16] and [55], a topology adaptation approach is used to minimize the distribution of inauthentic files by the malicious peers in P2P networks. However, these schemes do not work well in the unstructured networks. The unstructured P2P networks also suffer from the poor search scalability problem. The traditional mechanisms such as controlled flooding, random walker and topology evolution all lack scalability. Zhuge et al. have proposed a trust-based a probabilistic search algorithm called *P-walk* to improve the search efficiency and to reduce unnecessary traffic in P2P networks [68]. In P-walk, the neighboring peers assign trust scores to each other. During the routing process, the peers preferentially forward the queries to the highly ranked neighbors. However, the performance of the algorithm in large-scale unstructured networks is questionable. To combat the free-riders, various trust-based incentive mechanisms are presented in [57]. Most of these mechanisms, however, involve large overhead of computations.

To combat the problem of inauthentic downloads as well as to improve search scalability while protecting the privacy of the users, this chapter proposes an *adaptive trust-aware protocol* that is robust and scalable. The proposed protocol increases the search efficiency by suitably exploiting the semantic community structures formed as a result of topology adaptation, since most of the queries are resolved within the semantic communities. Moreover, it effectively combines the functionalities of a robust trust management model and the semantic community formation thereby making the searching process secure and efficient while protecting the privacy of the users. The trust management module uses direct observations by a peer about its neighbors as well as the indirect observations reported by neighbors about the peers in its neighborhood. The direct observations are referred to as the *first-hand information*, while the observations reported by the neighbors of a peer are referred to as the *second-hand information.* The trust management module computes the trust metrics for different peers, and based on the values of the trust metrics, it segregates the honest peers from malicious peers using both first-hand and second-hand information. The semantic community formation allows topology adaptation to form cluster of peers sharing similar contents in the network. The formation of the semantic communities also enables the protocol to form a neighborhood of trust which is utilized to protect user privacy in the network. The work presented in this chapter is an extension of our already published work [49] [50]. The specific contributions made in present work are: (i) the usefulness of the proposed protocol in the context of the IoT has been identified, (ii) an extensive state-of-the-art survey of the existing searching mechanisms for P2P networks has been presented so that the specific contributions of the proposed protocol can be understood clearly, (iii) the impact of the phenomena of node churning and free riders on the proposed protocol are analyzed, and (iv) a detailed comparative analysis of the proposed protocol with two existing protocols is presented.

The rest of the chapter is organized as follows. Section 2 discusses some of the existing search protocols for structured and unstructured P2P networks. Section 3 presents the proposed protocol for secure and privacy-aware searching. Section 4 introduces various metrics to measure the performance of the proposed protocol. In Section 5, we present the performance results of the protocol based on the metrics defined in Section 4. A brief discussion is also made on the comparative analysis of the performance of the proposed protocol with some of the existing similar protocols in the literature. Section 6 concludes the chapter while highlighting some future scope of work.

2 Related Work

In this section, we briefly describe some of the searching protocols for P2P networks existing in the literature. We broadly divide these protocols into three categories: (i) general searching schemes, (ii) secure searching schemes, and (iii) privacy-preserving searching schemes. The primary objective of the schemes under the general searching category is to enhance the search efficiency - i.e., to reduce the search time, to increase the scalability, and fault-tolerance etc. The secure searching schemes attempt to incorporate security into the searching mechanisms by defending against various possible attacks on the peers and the overall network. The privacy-preserving searching mechanisms protect peer (i.e., the user) privacy while carrying out the searching operation. In the following subsections, we briefly discuss some of the currently existing schemes under each of these three categories of search.

2.1 General Searching Schemes in P2P Networks

De Mello et al. have proposed a searching mechanism that is based on the discovery of trust paths among the peers in a P2P network [20]. Li & Wang proposed a global trust model based on the distance-weighted recommendations to quantify and evaluate the peers in a P2P network [36]. Adamic et al.[4] propose random-walk strategies in power-law networks(refer to Section 3.1), and find that by modifying the walkers to seek out for the peers having high degrees, the search performances in P2P networks can be greatly enhanced. However, such strategies lack scalability and do not perform well in a network having large number of peers.

Condie et al. presented a protocol named *adaptive peer-to-peer technologies* (APT) for the formation of adaptive topologies to reduce spurious file downloads and free riding. The peers connect to those peers from whom they are most likely to download the authentic files [16]. The peers add or remove their neighbors based on *local trust* and *connection trust* which are decided based on the transactions history. The scheme follows a defensive strategy for punishment since the peers follow the same strategy of punishment for both the malicious peers as well as the neighbors through whom they receive the responses from the malicious peers. This punishment strategy is relaxed in the *reciprocal capacity-based adaptive topology protocol* (RC-ATP), wherein a peer connects to others which have higher reciprocal capacities [55]. The *reciprocal capacity* of a peer is defined based on its capacity of providing good files and also on its ability of recommending the source of authentic files for download. While the RC-ATP scheme provides better network connectivity than the APT scheme and it also reduces the cost due to the inauthentic downloads, it has a large overhead due to the topology adaptation. Kamvar et al. proposed an algorithm that reduces the number of downloads of inauthentic files in a file-sharing P2P network [34]. Each peer is assigned a unique global trust value that is computed based on the historical activities of the peer in the network. A distributed and secure method based on *power iteration* is also presented for computing the global trust values of the peers. Based on the global trust values, the malicious peers are identified and isolated from the network. Xiao et al. have proposed an *adaptive connection* *establishment* (ACE) protocol that constructs an overlay multicast tree by including each source peer and the neighboring peers within a certain diameter from the source peer [60]. It further optimizes the connecting edges in the overlay graph that are not included in the tree while retaining the scope of the search. The protocol is fully distributed since the peers do not need global knowledge of the whole overlay network while using the search protocol.

In [61], which is known as Gnutella v0.6 system, a two-layer hierarchical structure is deployed. The peers are categorized into two types: the *leaf-peer* and the *ultra-peer*. The leaf-peers have connections with their respective ultra-peers, while the ultra-peers have connections with their own leaf-peers as well as with the other ultra-peers. The leaf-peers can initiate lookup requests, receive lookup responses and respond to requests for which they have exact answers. An ultra-peer forwards the lookup requests to other the ultra-peers or the leaf-peers to which the ultra-peer is connected, if it exactly knows which leaf-peer has answers to the requests. At the ultra-peer level of the hierarchy, a flooding mechanism is used for forwarding the lookup requests.

Hsiao et al. have addressed the *topology mismatch problem* in unstructured P2P networks using a novel topology matching algorithm [30]. In the proposed algorithm, each peer creates and maintains a constant number of overlay connections with other peers in a distributed manner. Tang et al. have proposed an analytical scheme that studied the search performance in a P2P network under time-to-live (TTL)-based search [56]. In [67], a fully distributed protocol named *distributed cycle minimization protocol*(DCMP) has been presented that minimizes duplicate messages by eliminating any possible redundant cycles of the messages. Lin el al. proposed a dynamic search algorithm which combines the strategies of flooding and *random walk* [38].

Li et al. have proposed a consistency maintenance scheme for heterogeneous P2P systems with shorter convergence time and light-weight bandwidth consumption by taking into consideration the network locality information and the heterogeneity of peer capacities in the network [37]. Martinez-Yelmo et al. have proposed a two-level hierarchical P2P overlay architecture for inter-connection of different *P2P session initiation protocol* (P2PSIP) clusters [40].

Zhang & Hu have presented a protocol for P2P search with the assistance from a partial indexing service based on the interests of the peers and the data popularity [66]. Yang & Yang proposed a two-level hybrid P2P system to make use of the advantages of both structured and unstructured P2P networks [65]. The upper level of the system is a structured core network which forms the backbone of the hybrid system while the lower level consists of multiple unstructured P2P networks each of which is attached to a super-peer at the upper level. Huang-Fu et al. proposed a hybrid P2P system for mobile devices that utilizes the short message service as the control protocol to identify the address of the called peer [31]. In the proposed scheme, the *mobile station integrated services digital network* (MSISDN) number, i.e., the telephone number, is used as the globally unique identification for each participating peer. Joung & Lin have proposed a fully decentralized algorithm to build a hybrid P2P system that does not need any human intervention and does not involve any centralized gateway to select the peers or to guide the peers to build a structured overlay [32].

Gkantsidis et al. [26] propose several hybrid search schemes for unstructured P2P networks. The authors have studied the performances of the search strategies in terms of several metrics such as: the number of distinct peers discovered, the number of messages propagated (i.e. communication overhead), and the maximum response time for the search queries. The authors have evaluated the performance of normalized flooding in non-regular P2P networks to show that normalization in flooding effectively tackles the problems caused by non-regularity in the network. It has also been shown that 1-step replication is helpful in search by random walk as well as search by normalized flooding, especially when the network has a small number of supernodes. The authors have utilized the theory of random graph to develop new algorithms based on *edge criticality heuristics* [35] used in the theory of approximate algorithms[58].

2.2 Secure Searching Schemes in P2P Networks

While efficiency of searching has been the major focus in most of the aforementioned schemes, security has also attracted attention of the researchers. Balfe et al. have discussed how the concepts of trusted computing can be applied to secure P2P networks [6]. The authors have argued that the central problem in securing P2P network lie in the fact that these networks do not have any stable verifiable peer identity verification mechanism. This leads to a major conflict between the requirements of anonymity of the users to protect their privacy and an increasing need to provide robust access control, data integrity, confidentiality and accountability services. The authors have shown how the *trusted computing group* (TCG) protocols for *direct anonymous attestation* (DAA) can be used to enforce the use of stable, platform-dependent pseudonyms so that spoofing attacks can be prevented. The proposed scheme also uses the DAA protocol to build entity authentication using pseudonyms for establishing secure communication channels between any given pair of peers.

Dingledine et al. [22] and Douceur [23] discuss various ways in which the spoofing attacks can be launched by malicious peers in a network that does not have a trusted central authority to verify the identities of the peers. The authors have proposed the use of reputation of the peers and the micro-cash schemes to detect such attacks. Sit & Morris [54] present a framework for performing a security analysis in P2P networks. The authors have proposed a taxonomy of attacks at the various layers of the communication protocol stack in the peers. At the network layer, attacks have been identified in the routing table lookup, maintenance, and route discovery process. The possible attacks on the file storage systems in the peers, and various types of *denial of service* (DoS) attacks on the peers and on the overall network have also been identified.

A large number of studies have been carried out on the reputation and trust management in both the unstructured and the structured P2P networks. The reputation schemes such as EigenTrust [34] and PeerTrust [63] have been proposed to work on top of the structured P2P networks such as CAN [41] and P-Grid [3]. Aberer & Despotovic have proposed a scheme to identify the dishonest peers based on the complaints received from the honest peers in the network [2]. However, since the scheme uses the negative feedbacks only, no distinction can be made between an honest peer and a peer which has not been active for some time or a newly joined peer. The EigenTrust scheme proposed by Kamvar et al. [34] evaluates the trust information provided by the peers based on their trustworthiness. The scheme utilizes a novel normalization process in which the trust ratings of a peer are averaged and normalized. However, the normalization may lead to partial loss of important information on the original distribution and variance of trust function.

In a scheme proposed by Damiani et al. [19], the trustworthiness of file is determined based on a voting mechanism invoked among the participating peers in a P2P system. However, the scheme does not distinguish between the votes of the peers having high reputation values from those of the peer with low reputation. Hence, peers having low reputation values can manipulate the final result of the vote thereby making the scheme unreliable. Xiong & Liu [62] propose a scheme for evaluating trust in the peers in a P2P e-commerce environment. Cha & Kim propose a reputation management scheme based on the unbiased collective intelligence of the nodes in a P2P network for identifying and removing fake multimedia files [12].

2.3 Privacy-Preserving Searching Schemes in P2P Networks

Since the protection of the privacy of the users has become a critical requirement over the years, the researchers have attempted to address this issue in P2P protocol designs. One easy way to preserve the privacy of the users in network communication is to deploy some fixed servers or proxies for this purpose. For example, in the Publius system [59], the identity of a publisher is protected by encrypting the data communicated in the network, and managing the key distribution among k servers by using the mechanism of *threshold cryptography* [21][52]. Some anonymity schemes based on the use of a trusted third party server have been presented in [64]. A scheme called "APES" has been proposed to achieve mutual anonymity in a peer-to-peer file sharing system [46].

One popular approach for preserving peer privacy in a P2P system is to reveal the identity of the previous peer only over an entire multi-hop route from a source to a destination. FreeNet [15][14], Crowds [43], Onion routing [28][42], and the shortcut responding protocol [64] are few examples of this approach. Although very effective for hiding the identity of the peers, this approach has a serious problem in P2P systems. In a P2P system, the logical neighbor of a peer may be far away in terms the physical distance. Multi-hop communications over such long links usually lead to high rate of packet drops, delay, and jitter.

Lu et al. propose a trust-based privacy preservation scheme for P2P data sharing [39]. The proposition is based on selection of a trusted peer as the proxy during the data acquirement. The requester peer sends the request and receives the data through the proxy without revealing its identity. Since the real identity of the requester is never revealed during the communication, the privacy of the requester node is protected. However, in an structured P2P network, the selection and maintenance of the trusted peers for each peer is difficult due to the dynamic nature of the network topology and the autonomy of the peers. Hence, the scheme is difficult to deploy in real-world networks.

In [53], a *peer-to-peer personal privacy protocol* (p^5) has been proposed for protecting the privacy of a sender-receiver pair. In this protocol, the packets from a source are transmitted to all the members of a broadcast group to which the source and the receiver belong, instead of sending the packets to the receiver only. To ensure confidentiality of the message, each packet is encrypted by the sender using the public key of the receiver. In order to maintain the traffic level in the network at a constant level, the peers generate noise packets if they have no real packets to send. The use of noise packets makes it impossible for an eavesdropper to distinguish a data packet form a noise packet. The anonymity is achieved at the cost of the suboptimal utilization of the network bandwidth.

Goel et al. have proposed a peer-to-peer communication system - named *Herbivore* - that can ensure provable anonymity of the peers [27]. The idea behind its design is borrowed from the well-known *dining cryptographer networks* [13]. To make the anonymization protocol scalable, the network is logically partitioned into a large number of small anonymizing cliques. For anonymizing one bit of information, Herbivore has to propagate at least $2(k$ - $1)$ bits, where k is the size of the clique in which the message is being communicated. Moreover, if a node has to send a packet, for achieving anonymity, all the other peers in the same clique will have to send at least the same amount of data. This results in a high communication overhead in the protocol.

The Motivation of the Proposed Protocol: The protocol presented in this chapter draws its motivation from the APT [16] and RC-ATP [55] protocols. However, there are some significant differences between the protocol presented in this chapter and the APT and the RC-ATP protocol. First, in the proposed protocol, the links in the original overlays are never deleted in order to avoid network partitioning. Second, in presence of malicious peers, the robustness of the proposed protocol is higher than that of the APT and the RC-ATP protocol. This claim is validated by the simulation results presented in Section 5. Third, as APT and RC-ATP both use flooding to locate resources, they have the typical problem of scalability in searching. The protocol presented in this chapter takes the advantage of semantic communities formation to improve the *quality of service* (QoS) of search by reducing the search time and increasing the rate of authentic file downloads. Fourth, APT and RC-ATP do not employ any robust trust model for ensuring security in searching and for protecting the user identity and data privacy. On the other hand, the central module of the proposed protocol is a robust trust management framework, which is responsible for securing the searching process and protecting the privacy of the peers and their data. Finally, unlike the APT and the RC-ATP protocols, the proposed protocol punishes the malicious peers by blocking all the queries which originate from these peers. This ensures that the malicious peers are not allowed to consume the resources and the services available in the network.

3 The Secure and Privacy-Aware Searching Protocol

This section is divided into three sub-sections. In Section 3.1, various parameters and the network environment of P2P network for which the proposed protocol is designed are discussed. In Section 3.2, the proposed search protocol is presented. Finally, Section 3.3 describes how the user privacy is protected in the proposed searching protocol.

3.1 The Network Environment

To obtain reliable results, the proposed protocol is evaluated on a realistic P2P network model. The factors that are taken into consideration in designing the protocol are discussed below.

(1) Network Topology: The topology of a P2P network plays an important role in the formation of trust among its peers and for efficient operation of a search protocol in the network. Following the work in [16] and [55], in the current proposition, the P2P network has been modeled as a *power law graph*. In a power law network, the degree distribution of the peers follows a *power law distribution*, in which the fraction of peers having degree L is L^{-k} where k is a network dependent constant. In the network environment, a certain percentage of the peers are randomly chosen to act as malicious peers. The malicious peers distribute bogus files in the network. As the protocol executes, the peers adjust topology locally to connect to those peers which have better chance to provide good files in future, and drop the malicious peers from their neighborhood. The network links are categorized into two types: *connectivity link* and *community link*. The connectivity links are the edges of the original power law network which provide seamless connectivity among the peers. To prevent the network from being partitioned, these links are never deleted. On the other hand, the community links are added probabilistically between the peers who know each other, and have already interacted with each other before. A community link may be deleted when the perceived trustworthiness of a peer falls in the perception of its neighbors. The formal procedure of computing trust of a peer is discussed later in this section. However, informally, it may be said that the value of the trust metric of a peer i as computed by another peer j increases when the peer j has some positive experience while interacting with the peer i (i.e. getting an authentic file from peer the i). A negative experience (i.e. getting a bogus file) leads to decrease in the trust value. A limit is put on the additional number of edges that a peer can acquire to control the bandwidth usage and the query processing overhead in the network. This increase in network load is measured relative to the initial network degree (corresponding to the connectivity edges). Let *final_degree(x)* and *initial_degree(x)* be the initial and the final degree of a node x. The *relative increase in connectivity* (RIC) as computed in (1) is constrained by a parameter called *edge_limit*.

$$RIC(x) = \frac{final\_degree(x)}{initial\_degree(x)} \leq edge\_limit \tag{1}$$

(2) Content Distribution: The dynamics of a P2P network are highly dependent on the volume and the variety of the files that each peer chooses to share. Hence a model reflecting the real-world P2P networks is required. It has been observed that the peers are, in general, interested in a subset of the contents in the P2P network [17]. Also, the peers are often interested only in the files from a few content categories. Some categories of files are more popular than the others. It has been shown that the Gnutella content distribution follows the *zipf distribution* [48]. In the proposed scheme, the files are assigned to the peers at the network initialization phase as follows: The peer i is assigned some content categories C^i and the peer i is given an interest level for each content category $c \epsilon C$. Finally, the peer i is assigned files F according to its content categories and interest levels in those categories. Each distinct file $f_{(c,r)}$ is uniquely identified by the content category c to which it belongs and its popularity ranking r within that category [48].

Accordingly, in the proposed protocol, the content categories and the file popularity within each category are both modeled as *zipf distribution* with $\alpha = 0.8$.

Content distribution model: In the proposed scheme, we assume that there are 32 content categories. It must be noted that the number of content categories can be any positive integer n. However, for the evaluation of the performance of the proposed protocol, we have used 32 content categories.

Table 1 An illustrative content distribution among peers

Peers	**Content Categories**
P_1	C_1, C_2, C_3
P_2	C_3, C_4, C_6, C_7
P_3	C_2, C_4, C_7, C_8
P_4	C_1, C_2
P_5	C_1, C_5, C_6

Let the content categories be $C = \{c_1, c_2,..., c_{32}\}$. Each content category is characterized by its popularity rank. For example, if $c_1 = 1$, $c_2 = 2$ and $c_3 = 3$, then c_1 is more popular than c_2 and hence it is more replicated than c_2 and so on. Since, the files in the more popular categories are searched and queried more frequently, more number of these files are stored and replicated than the files belonging to the less popular categories. This strategy ensures that more number of files belonging to the popular categories are available, which, in turn, makes the searching process faster and efficient. More details on the way in which the content categories are assigned to the peers and the interest levels of the peers are modeled can be found in [48].

As already discussed earlier, the peers are assumed to be interested in a subset of the total available contents in the network. Accordingly, each peer initially chooses a number of content categories and shares files only in those categories. In the proposed protocol, each peer randomly chooses between three to six content categories. The files belonging to more popular categories are shared more in numbers. Table 1 shows an illustrative content distribution among 5 peers in a network. The category C_1 is more replicated as it is the most popular category. Peer 1 (P_1) shares files of three categories: C_1, C_2, C_3. As explained earlier, P_1 shares maximum number of files in category C_1, followed by category C_2 and so on. On the other hand, Peer 3 (P_3) shares maximum number of files in category C_2 as it is the most popular among the categories of files chosen by it.

(3) Query Initiation Model: The authors in [48] have shown that the peers usually query for the files which are available in the network, and which belong to the content categories of their interests. However, the number of queries a peer issues may vary from peer to peer. Using the *Poisson* distribution this is modeled as follows: If M is the total number of queries issued in a cycle, and N is the number of peers present in the network, query rate $\lambda = M$ / N is the mean of the Poisson process. The expression: $p(\# of queries = K)$ $= \frac{e^{-K}\lambda^{K}}{K!}$ gives the probability that a peer issues K queries in a cycle. The probability that a peer issues a query for the file $f_{c,r}$ depends on the peer's interest level in category c and rank r of the file within that category.

When a peer generates a query, instead of generating a search string, it generates the category and the rank (i.e., popularity) of the file that will satisfy the query. On receiving the query, each peer checks whether it supports the file category and if so, whether it shares the file.

(4) Trust Management Engine: A trust management engine is designed which helps a peer to compute the trust ratings of the other peers based on the past transactions, as well as on the recommendations of its neighbor. For computing the trust values of the peers, a method similar to the one proposed in [24] is followed. The framework employs a *beta distribution* for reputation representation, updates and integration. The first-hand information and the second-hand information (recommendation from neighbors) are combined to compute the reputation value of a peer. The weight assigned by a peer i to a second-hand information received from a peer k is a function of the reputation of the peer k as maintained in the peer i. For each peer j, a reputation R_{ij} is computed by a neighbor peer i. The reputation is embodied in the *Beta model* which has two parameters: α_{ij} and β_{ij}. α_{ij} represents the number of successful transactions (i.e., the number of authentic file downloads) that the peer i had with the peer j, and β_{ij} represents the number of unsuccessful transactions (i.e., the number of unauthentic file downloads). The reputation of the peer j as maintained by the peer i is computed using (2).

$$R_{ij} = Beta(\alpha_{ij} + 1, \beta_{ij} + 1) \tag{2}$$

The trust metric of a peer is the expected value of its reputation and is given by (3).

$$T_{ij} = E(R_{ij}) = E(Beta(\alpha_{ij} + 1, \beta_{ij} + 1)) = \frac{\alpha_{ij} + 1}{\alpha_{ij} + \beta_{ij} + 2} \tag{3}$$

The second-hand information is presented to the peer i by its neighbor peer k. The peer i receives the reputation R_{kj} of the peer j from the peer k in the form of the two parameters α_{kj} and β_{kj}. After receiving this new information, the peer i combines it with its current assessment R_{ij} to obtain a new reputation R_{ij}^{new} as shown in (4).

$$R_{ij}^{new} = Beta(\alpha_{ij}^{new}, \beta_{ij}^{new}) \tag{4}$$

In (4), the values of α_{ij}^{new} and β_{ij}^{new} are given by (5) and (6) as follows.

$$\alpha_{ij}^{new} = \alpha_{ij} + \frac{2\alpha_{ik}\alpha_{kj}}{(\beta_{ik} + 2)(\alpha_{kj} + \beta_{kj} + 2)(2\alpha_{ik})} \tag{5}$$

$$\beta_{ij}^{new} = \beta_{ij} + \frac{2\alpha_{ik}\beta_{kj}}{(\beta_{ik} + 2)(\alpha_{kj} + \beta_{kj} + 2)(2\alpha_{ik})} \tag{6}$$

To prevent against *bad-mouthing* and *ballot-stuffing attacks* [10][18], the peers assign higher weights to the first-hand observations (i.e. direct observations) made by them, and less weights are given to the evidences provided by the other peers (i.e., the second-hand information). As mentioned earlier in this section, the second-hand observations received from different peers are also weighted in proportions to the values of their respective reputation metrics. To incorporate these issues while updating the reputation values using the second-hand information, the *Dempster-Shafer theory* [51] and the *belief discounting model* [33] are employed. The use of these models leads to the derivation of the expressions in (5) and (6).

To make the trust management system robust against *sleeper attack* [25], where a peer behaves honestly for a sufficiently long time to acquire a good reputation and then starts misbehaving and exploiting the system, the proposed system assigns more weights to the recent observations for computing the aggregate reputation metrics of a peer. In this approach, the reputation metrics of a peer are periodically decreased by a weight w, using (7) and (8).

$$\alpha_{ij}^{new} = w * \alpha_{ij} \tag{7}$$

$$\beta_{ij}^{new} = w * \beta_{ij} \tag{8}$$

The choice of the weight w in (7) and (8) and the interval at which the reputation updates are made are two *tuneable parameters*. In [11], a technique has been proposed for computing the weight(w) by comparing the reputation evolution in the system with and without the weighting parameters.

As mentioned earlier in this section, the trust value of a peer is computed as the statistical expected value of its reputation. The trust value of a peer lies in the interval [0, 1]. The peer i considers the peer j as trustworthy if $S_{ij} \geq 0.5$, and malicious if $S_{ij} < 0.5$. In the implementation of the proposed protocol, we have used an LRU (least recently used) data structure which is maintained in each peer to keep track of the most recent transactions the peer had with maximum of 32 peers. However, the choice of the number of peers whose transaction history is maintained in each peer is a tuneable parameter, which can be increased or decreased based on the memory and the computing capabilities of the peers.

(5) Identity of the Peers: Each peer generates a 1024 bit public/private RSA key pair. The public key serves as the identity of the peer. The identities are persistent and they enable two peers that have exchanged keys to locate and connect to one another whenever the peers are online. In addition, a *distributed hash table* (DHT) is maintained that lists the transient IP addresses and the port numbers for all the peers and for all the applications running on the peers. The DHT entries for the peer i are signed by the peer i and encrypted using its public key. Each entry is indexed by a 20 byte randomly generated shared secret, which is agreed upon during the first successful connection between the two peers. Each peer's location in the DHT is independent of its identity and is determined by hashing the client's current IP address and the DHT port. This prevents any possible systematic monitoring of the targeted regions of the DHT key space, since the region for which each peer is responsible is determined by the peer's network address and the port.

(6) Node Churning Model: In P2P networks, a large number of peers may join and leave at any time. This activity is termed as *node churning.* To simulate node churning, prior to each *generation* (a set of consecutive searches), a fixed percentage of nodes are chosen randomly as *inactive peers.* These peers neither initiate nor respond to a query in that generation, and they join the system later with their LRU data structure cleared. The clearing of the LRU data structure ensures that these peers do not have any historical information about their past transactions with other peers in the network. Since in a real world network, even in presence of churning, the approximate distribution of content categories and files remain constant, the contents of the peers undergoing churn are exchanged with the peer remaining in the network, so that the content distribution model of the network remains unchanged.
(7) Threat Model: The malicious peers adopt various strategies (threat models) to conceal their behavior so that they can effectively disrupt the activities in the network, and yet go undetected. The proposed protocol considers two threat models. The peers which share good quality files enjoy better topological positions after topology adaptation. In the threat model A, the malicious peers attempt to circumvent this effect by providing good files (occasionally) with a probability - known as *degree of deception*- to lure other peers to form communities with them. In the threat model B, a group of malicious peer joins the system and provides good files until the connectivity of the peers reaches a maximum value - the *edge limit.* The peers then start acting maliciously by spreading fake contents in the network. In Section 5, we will see how effective these strategies are in disrupting the network operations.

3.2 The Proposed Search Protocol

The network learns the trust information through the search process, and updates the trust information and adapts then topology based on the outcome of the search. An ideal search protocol should satisfy several requirements such as: (a) It should have a high search efficiency and search quality - i.e. it must have the ability to download authentic files in a short period of time, (b) it should have a minimal overhead in terms of computation, storage and message passing, (c) It must provide incentives to the peers which share a large number of authentic files, (d) it should be self-policing in the sense that a peer should be able adjust its search strategy based on the local estimate of the network connectivity, and (e) it should be able to protect the privacy of its users. The proposed search protocol has been designed to satisfy each of these requirements.

The proposed protocol works in three steps: (i) search, (ii) trust computing and verification, and (iii) topology adaptation. Each of these steps is discussed in the following.

Search: A *time to live* (TTL) bound search is used. At each peer, the query is forwarded to a subset of its neighbors; the number of neighbors is decided based on the local estimate of connectivity. The *connectivity index* for the peer x is denoted as $Prob_{com}(x)$ and is given by (9).

$$Prob_{comm}(x) = \frac{current\_degree(x) - initial\_degree(x)}{initial\_degree(x)(edge\_limit - 1)} \tag{9}$$

When $Prob_{com}$ for a node is low, the peer has the capacity to accept new community edges for expanding the community structure. Higher the value of $Prob_{com}$, it is less likely that the neighbors will disseminate the queries. As the protocol executes, the connectivity of the good peers increases and finally reaches a maximum value. At this time, the peers focus on directing the queries to appropriate communities which may host the specific file rather than expanding the communities. For example, if peer i can contact at most 10 neighbors and $Prob_{com}$ of i is 0.6, it forwards the query to: 10 x (1 - 0.6) = 4 neighbors only. The search strategy is changed from the initial TTL-limited *breadth first search* (BFS) to a directed *depth first search* (DFS) with the restructuring of the network. The search process operates in two steps: *query initiation* and *query forward*. These steps are described in the following.

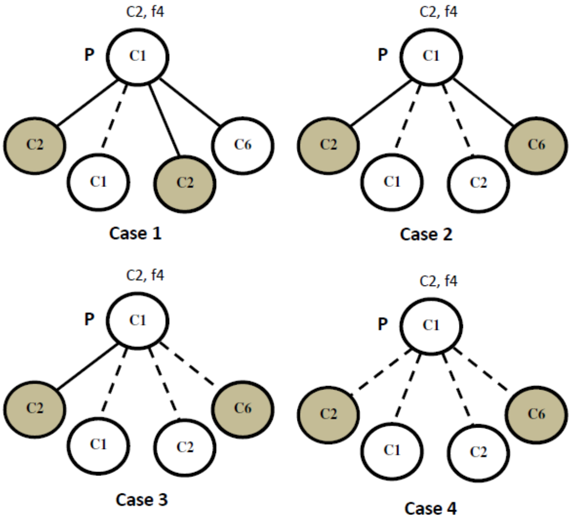


Fig. 1 Neighbor selection by peer P for forwarding the query string (c_2, f_4). The community edges and the connectivity edges are drawn using solid and dotted lines respectively. The peers that receive the query for forwarding are shaded.

(i) Query Initiation: The initiating peer forms a query packet containing the name of the file (c, r) and forwards it to some of its neighbors along with the $Prob_{com}$ and the TTL values. The query is disseminated using the following *neighbor selection rule*. The neighbors are ranked based on both their trustworthiness and their similarities of interest. Preference is given to the trusted neighbors sharing similar contents. Among the trusted neighbors, the community members having their contents matched to the query are preferred. If the number of community links is not adequate enough, the query is forwarded through the connectivity links also. The various cases of neighbor selection are illustrated in Fig. 1. It is assumed that in each case only two neighbors are selected for forwarding a query. When the query (c_2, f_4) reaches the peer P, following four cases may occur. In Case 1, the peer P has sufficient number of community neighbors (two community neighbors) sharing files in the category c_2. Hence, these peers are chosen for forwarding the query. In Case 2, the number of community neighbors sharing the requested category of file is not sufficient enough - only one community neighbor has the file in the category c_2. In this scenario, the community neighbors sharing the c_2 and the c_6 categories of files are preferred over the connectivity neighbor sharing the file category c_2 for forwarding the query. This is because of the fact that the peers forward queries to the community peers which have higher trust values than the connectivity peers. In Case 3, there is only one community neighbor that shares the file category c_2. Hence that neighbor is chosen for the purpose of query forwarding. Among the remaining connectivity neighbors, the most trusted one containing the c_6 category is selected. In Case 4, there are no community neighbors. Assuming that the peer P has the same level of trust for all its neighbors, the neighbor sharing the matching content category c_2 is chosen for forwarding the query. Among the rest of the neighbors, the peer c_6 is chosen randomly (since only two forwarding peers are to be selected).

When a query reaches peer i from peer j, peer i forwards the query further in the network as discussed below.

(ii) Query Forwarding: (i) *Check the trust level of the peer* j: The peer i checks the trust rating of the peer j through the *check trust rating* algorithm (explained later in this section). The selection of the peers for further forwarding of the query is done accordingly. (ii) *Check the availability of the file*: If the requested file is found, a response is sent to the peer j. If the TTL value has not expired, the following steps are executed. (iii) *Calculate the number of messages to be sent*: The number of messages to be sent is calculated based on the value of $Prob_{com}$.(iv) *Choose the neighbors*: The neighbors are chosen using the neighbor selection rule. The search process is shown in Fig. 2. It is assumed that from each peer, the query is forwarded to two neighbors. The matching community links are preferred over the connectivity links to dispatch the query. The peer 1 initiates the query and forwards it to two community neighbors 3 and 4. The query reaches the peer 8 via the peer 4. However, the peer 8 knows from its previous transactions with the peer 4 that the peer 4 is malicious. Hence, it blocks the query. The query forwarded by the peer 5 is also blocked by the peer 10 and the peer 11 as both of them know that the peer 5 is malicious. The query is matched at four peers: 4, 6, 9 and 12. The search process is shown in Fig. 2.

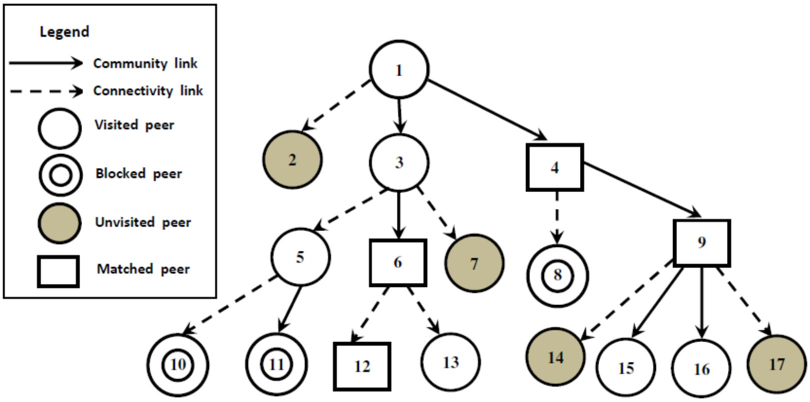


Fig. 2 The breadth first search (BFS) tree for the search initiated by peer 1

Topology Adaptation: The responses are sorted by the initiating peer i based on the reputations of the resource providers, and the peer having the highest reputation is selected as the source for downloading. The requesting peer checks the authenticity of the downloaded file. If the file is found to be fake, the peer i attempts to download the file from other sources until it is able to find the authentic resource or it does have any sources left for searching. The peer then updates the trust ratings and possibly adapts the network topology after a failed or a successful download, to bring the trusted peers closer to its neighborhood, and to drop the malicious peers from its community. The restructuring of the network is controlled by a parameter known as *degree of rewiring* which represents the probability with which a link is formed between a pair of peers. This parameter allows the trust information to propagate through the network. The topology adaptation consists of the following operations: (i) *link deletion*: The peer i deletes the existing community link with the peer j if it detects the peer j as malicious. (ii) *link addition*: The peer i probabilistically forms a community link with the peer j if the resource provided by the peer j is found to be authentic. If $RIC \leq edge_{limit}$, for both the peers i and j, only then an edge can be added, subject to the approval of the resource provider peer j. If the peer j finds that the peer i is malicious (i.e., its trust value is below the threshold), it doesn't approve the link.

Fig. 3 illustrates a topology adaptation on the network topology shown in Fig. 2. In the example shown in Fig. 3, the peer 1 downloads the file from the peer 4 and finds that the file is spurious. It reduces the trust score of the peer 4 and deletes the community link 1-4. It then downloads the file from

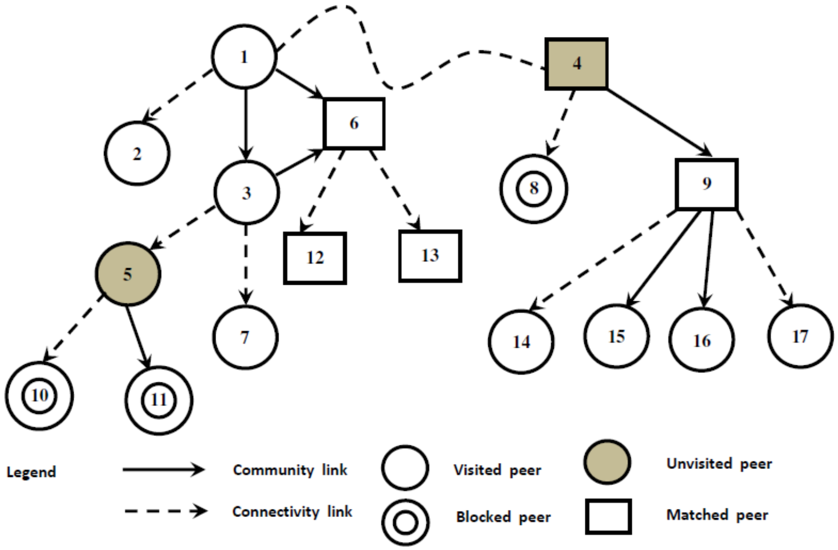


Fig. 3 Topology adaptation based on outcome of the search in Fig. 2. Malicious nodes are shaded in gray color.

the peer 6 and gets an authentic file. The peer 1 now sends a request to the peer 6, and the latter grants the request after checking its trust value. Hence, the community edge 1-6 is added. The malicious peer 4 loses one community link and the peer 6 gains one community edge. However, the network still remains connected by the connectivity edges which are shown in dotted lines in Fig. 3.

It may be noted that the addition of a community link is a more expensive operation than the deletion of a community link. However, if the number of malicious peers in a network is not too high, the link addition operation will be less frequent after the formation of semantic communities and stabilization of the topology adaptation. Hence, except during the initial semantic community formation phase, the overhead of the protocol operation will never be high. This will be discussed in more detail in Section 5.

Checking of the Trust Rating of the Peers: The trust rating of the peers is used at various stages of execution of the protocol to make a decision on the possible source for downloading a file, to stop a query forwarded from a malicious node and to adapt the topology. A *least recently used* (LRU) data structure is used at each peer to keep track of the 32 most recent peers it has interacted with. When no transaction history is available, a peer seeks for the recommendations from its neighbors using a *trust query* message. When the peer i doesn't have the trust score of the peer j in its LRU history, it first seeks for the recommendation about the peer j from all of its community neighbors.

If none of its community neighbors possesses any information about the peer j, then the peer i initiates a *directed DFS search*. The trust computation model has been presented in Section 3.1.

3.3 Privacy-Preservation in Searching

The trust-based searching protocol described above does not guarantee any privacy requirement of the requester (i.e. the initiator of the query). For protecting the privacy of the user, several enhancement of the protocol are proposed. Following cases are identified for privacy preservation.

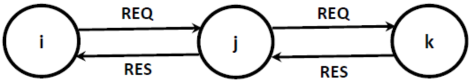


Fig. 4 Identity protection of the requesting peer i from the supplier peer k by use of trusted peer j. *REQ* and *RES* are the request and response message respectively.

(a) Protection of the Identity of the Requesting Peer: In this case, as shown in Fig. 4, instead of sending the request straightway to the supplier peer, the requesting peer asks one of its trusted peers (which may or may not be its neighbor) to look up the data on its behalf. Once the query propagation module successfully identifies the possible supplier of the resource, the trusted peer serves as a proxy to deliver the data to the requester peer. Other peers including the supplier of the resource will not be able to know the real requester. Hence, the requester's privacy is protected. Since the requestor's identity is only known to its trusted peer, the strength of privacy is dependent on the effort required to compromise the trusted peer. As mentioned in Section 3.1, the message communicated by the peers are encrypted by 1024 bit RSA key, which is a provably secure algorithm. Hence, the privacy of the requester peer is protected.

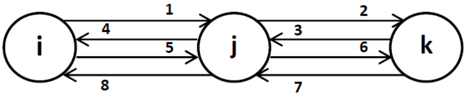


Fig. 5 Protecting data handle using trusted node. Peer i and k are the requester and the supplier peer respectively. Peer j is the trusted peer of the requester peer i.

(b) Protecting the Data Handle: To improve the achieved privacy level, the data handle may not be put in the request at the beginning. When a requester initiates the request, it computes the hash value of the handle and reveals only a part of the hash result in the request sent to its trusted peer. The steps 1 and 2 in Fig. 5 represent these activities. Each peer receiving the request compares the revealed partial hash to the hash codes of the data handles that it holds. Depending on the length of the revealed part, the receiving peer may find multiple matches. This does not, however, imply that the peer has the requested data. Thus this peer will provide a candidate set, along with a certificate of its public key, to the requester. If the matched set is not empty, the peer constructs a *Bloom filter* [9] based on the left parts of the matched hash codes, and sends it back to the trusted peer. The trusted peer forwards it back to the requester. These are represented by the steps 3 and 4 in Fig. 5. On examining the filters, the requester can eliminate all peers that do not have the required data from the candidate data supplier list. It then encrypts the complete request with the supplier's public key and gets the requested data with the help from its trusted peer. The steps 5, 6, 7 and 8 in Fig. 5 represent these activities. By adjusting the length of the revealed hash code, the requestor can control the number of eliminated peers. The level of privacy is improved manifold since the malicious peers now need to compromise the trusted peer and also break the Bloom filter and the hash function in order to attack the privacy protection scheme.

(c) Hiding the Data Content: Although the privacy-preservation level has been improved during the lookup phase using the previous two schemes, the privacy of the requester will still be compromised if the trusted peer can see the data content when it relays the packets for the requester. To improve the privacy level and prevent eavesdropping, we can encrypt the data handle and the data content. If the identity of the supplier is known to the requester, it can encrypt the request using the supplier's public key. The public key of the requester cannot be used because the certificate will reveal its identity. The problem is solved in the following manner. The requester generates a symmetric key and encrypts it using the supplier's public key. Only the supplier can recover the key and use it to encrypt the data. To prevent the trusted peer of the requester from conducting a man-in-the-middle attack, the trusted peer is required to sign the packet. This provides a non-repudiation evidence, and shows that the packet is not generated by the trusted peer itself. The privacy level has been improved, since now in order launch an attack on the privacy of the requester, a malicious peer needs to break the encryption keys as well.

4 Performance Metrics for the Proposed Protocol

To analyze the performance of the proposed protocol, several metrics are defined. In this section, we provide a detailed discussion on these metrics which are used to evaluate the protocol performance. The performance results of the protocol based on these metrics are presented in Section 5.

(a) Attempt Ratio (AR): A peer keeps on downloading files from various sources based on their trust ratings till it gets the authentic file. AR is the probability that the authentic file is downloaded in the first attempt. A high value of AR for the honest peers is desirable for a searching scheme to be efficient and scalable.

(b) Effective Attempt Ratio (EAR): It measures the cost of downloading an authentic file by a good peer in comparison to the cost incurred by a malicious peer. If $P(i)$ be the total number of attempts made by the peer i to download an authentic file, EAR is given by (10).

$$EAR = (\frac{1}{M}\sum_{i=1}^{M}\frac{1}{P(i)} - \frac{1}{N}\sum_{j=1}^{N}\frac{1}{P(j)}) \tag{10}$$

In (10), M and N are the number of malicious and good peers issuing queries in a particular generation. For example, EAR = 50 implies that if a good peer needs one attempt to download an authentic file, a malicious peer will need two attempts.

(c) Query Miss Ratio (QMR): Since the formation of semantic communities takes some time, there will be a high rate of query misses in the first few generations of search. However, as the protocol executes, the rate of query miss is expected to fall for the good peers. QMR is defined as the ratio of the number of search failures to the total number of searches in a generation.

(d) Hit per Message (HM): Due to the formation of the semantic communities in the network, the number of messages required to get a hit is expected to fall down as the network topology stabilizes. HM measures the search efficiency achieved by the proposed search protocol and it is defined as the number of query hits per message irrespective of the authenticity of the file being downloaded.

(e) Relative Increase in Connectivity (RIC): After a successful download, a requesting peer attempts to establish a community edge with the resource provider, if it is approved by the latter. This ensures that the peers which provide good community services are rewarded by providing them with an increased number of community neighbors. The metric RIC measures the number of community neighbors a peer gains with respect to its connectivity neighbors in the initial network topology. If $D_{init}(i)$ and $D_{final}(i)$ are the initial and the final degrees of the peer i, and N is the number of peers, then RIC for the peer i is computed using (11).

$$RIC(i) = \frac{1}{N}\sum_{i}\frac{D_{final}(i)}{D_{init}(i)} \tag{11}$$

(f) Closeness Centrality (CCen): Since the topology adaptation effectively brings the good peers closer to each other, the length of the shortest path between a pair of good peers decreases. This intrinsic incentive for sharing authentic files is measured by the metric CCen. The peers with higher

CCen values are topologically better positioned. If P_{ij} is the length of the shortest path between the peer i and the peer j through the community edges and if V denotes the set of peers, then CCen for the peer i is given by (12).

$$CCen(i) = \frac{1}{\sum_{j \varepsilon V} P_{ij}} \tag{12}$$

(g) Clustering Coefficient (CC): It gives an indication about how well the network forms cliques. CC plays an important role in the choice of the TTL value in the search protocol. With higher values of CC, lower TTL values can be used in the search operation. If K_i be the number of community neighbors of the peer i, then the CC of the peer i is computed using (13).

$$CC(i) = \frac{2E_i}{K_i(K_i - 1)} \tag{13}$$

In (13), E_i is the actual number of community edges between the K_i neighbors. CC of the network is taken as the average value of all CC(i)s.

(h) Largest Connected Component (LCC): The community edges connect the peers which have similar content interests and have sufficiently high mutual trust among each other. If we focus on the peers which share a particular category of contents, then we can observe that the community edges form a trust-aware overlay. However, it will be highly probable that the trust-aware overlay graph will be a disconnected graph. LCC is the largest connected component of this disconnected overlay graph. In other words, LCC of the network can be taken as a measure of the goodness of the community structure, since it signifies how strongly the peers with similar contents and interests are connected with each other.

(i) Trust Query Propagation Overhead (TQPO): The peers build trust and reputation information by collecting and using both the first-hand and the second-hand information. A trust query message is propagated when the trust information about a peer is not available locally in a peer. A trust query message involves one DFS round without any backtracking. The overhead incurred due to the trust query propagation is measured by the metric called *trust query propagation overhead* (TQPO). TQPO is defined as the total number of distinct DFS search attempts per generation. It may be noted that a trust query may be initiated multiple number of times for a single file search operation - to select a trusted neighbor or to approve a community link.

(j) Topology Adaptation Overhead (TAO): It gives an idea about the overhead due to the topology adaptation and it is measured by the number of community edges that are added or deleted in one cycle of operation of the search protocol. The larger the number of addition and deletion of the community edges, higher will be the associated overhead.

5 Performance Evaluation of the Proposed Protocol

A discrete time simulator written in C is used for simulating the protocol. In the simulation, 6000 peers, 18000 *connectivity edges*, 32 *content categories* are chosen. The values of the *degree of deception* and the *degree of rewiring* are taken as 0.1 and 0.3 respectively. The *edge_limit* value used is 2.5. The TTL values for the BFS and the DFS are taken as 5 and 10 respectively. Since one of the objectives of the simulation is to show higher scalability of the proposed protocol, the number of peers and the number of connectivity edges in the simulated network are chosen to much higher than those used in simulating the APT [16] and the RC-ATP [55] protocols, while the TTL value for BFS is kept constant. The values of the simulation parameters are presented in Table 2.

Table 2 Simulation parameters

Parameters	Values
No. of peers	6000
No. of connectivity edges	18000
No. of content categories	32
Degree of deception	0.1
Degree of rewiring	0.3
Edge limit	2.5
TTL for BFS	$5s$
TTL for DFS	$10s$
No. of search per generation	5000
No. of generations per cycle	100

The discrete time simulator simulates the protocol repeatedly on the power law network and outputs all the metrics averaged over the generations. *Barabasi-Albert* generator [8] is used to generate initial power law graph with 6000 nodes and approximately 18000 edges. The number of search per generation is taken as 5000 while the number of generations per cycle of simulation is 100.

To check the robustness of the protocol against attacks from malicious peers, the percentage of malicious peers is gradually increased. Fig. 6 illustrates the cost incurred by each type of peers to download the authentic files. It can be observed from Fig. 6(a) and Fig. 6(b) that with the increase in the percentage of the malicious peers in the network from 10% to 20%, the AR for the malicious nodes increases while the AR for the honest peers falls marginally. Since AR indicates the cost (in terms of number of attempts required for downloading an authentic file) incurred by a peer, it can be concluded that as the percentage of the malicious peers is increased, the cost incurred by the malicious peers to download the authentic files decreases while that of the good peers increases.

It is also observed from Fig. 7 that the EAR values for the peers decrease as the percentage of the malicious peers in the network is gradually increased from 10% to 60%.

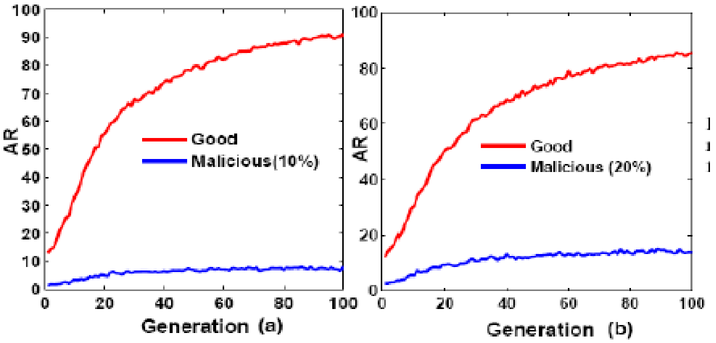


Fig. 6 AR for various percentages of malicious peers in the network. In (a) 10%, in (b) 20% nodes are malicious.

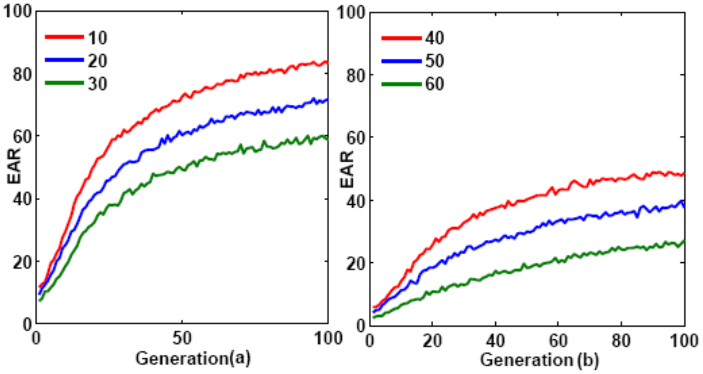


Fig. 7 EAR of honest peers for various percentages of malicious peers in the network. In (a) 10% - 30%, in (b) 40% - 60% peers in the network are malicious.

It is evident from Fig. 7 that when 10% of the peers in the network are malicious, the average EAR is 80; i.e., on the average, if a good peer needs one attempt to download an authentic file, a malicious peer needs 5 attempts. The peers which share high quality files acquire good reputation and earn more community edges and eventually disseminate the query through the community edges only. As the queries are forwarded via the trusted peers at each hop, the probability of getting the authentic files in the first attempt increases. However, as the queries forwarded by the malicious peers are blocked by the good peers, they need more attempts to download the good files. It may be observed from Fig. 7(b) that when the percentage of the malicious peers in the network is 60%, the value of EAR drops to 30. Hence, as long as the percentage of the malicious peers in the network does not exceed 60%, the good peers have higher probability to get the authentic files in their first attempts as compared to the malicious peers. The results, therefore, indicate that the proposed protocol can withstand attacks by the malicious peers till such peers are less than 60% of the total number of peers in the network.

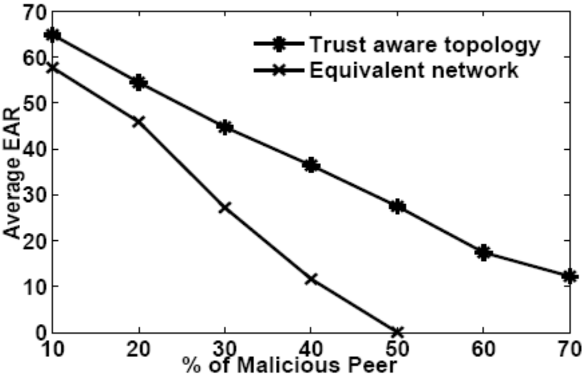


Fig. 8 Avg. EAR for various percentages of malicious peers in the network with and without the trust management module

The performance of the proposed protocol is compared with an equivalent power law network with no trust management framework in place. Since the proposed protocol allows addition of the community edges, therefore, to keep the number of edges in both the networks equal, additional edges are introduced between the similar peers in the equivalent network. Fig. 8 shows the comparison of the average EAR values. In the network without trust and reputation management, the value of EAR drops to zero when 50% or more of the peers in the network are malicious. However, in the network with the proposed protocol in place, even when 60% peers in the network are malicious, the value of EAR is consistently sustained at 20. This clearly demonstrates the robustness of the proposed protocol.

Fig. 9 shows the QMR experienced by both the types of peers for varying percentages of the malicious peers in the network. Initially, the value of QMR is high as no interest-based communities are formed and the searching is essentially a blind (i.e., brute force) one. As the protocol executes further, the peers with similar content interests come closer to each other (in terms

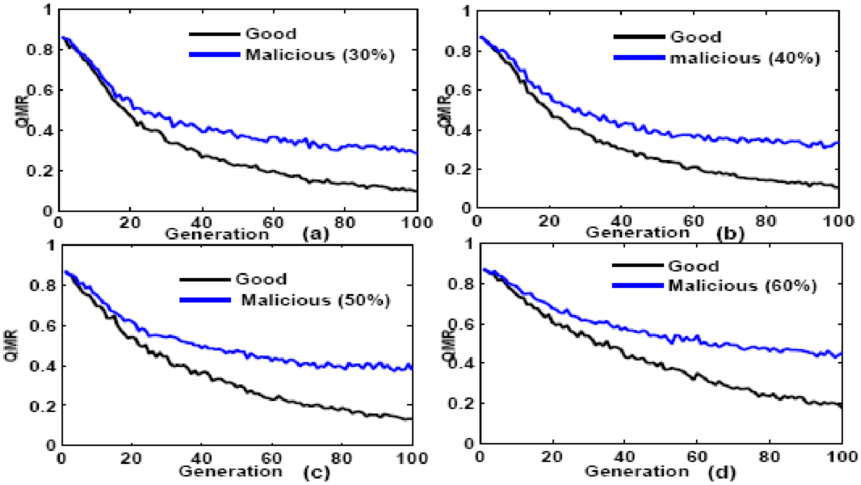


Fig. 9 QMR for various percentages of malicious peers in the network

of number of hops between them), and the queries are forwarded through the community edges. As a result, the value of QMR drops for the good peers. It is observed from Fig. 9 that the steady state value of the QMR for the good peers is less than 0.2, and the value of QMR is independent of the percentage of the malicious peers in the network. This is a significant performance achievement of the proposed protocol. For the malicious peers, the steady state value of QMR is 0.4. The high value of QMR for the malicious peers is due to the fact that the queries from the malicious peers are blocked by the good peers. It is evidently clear from the results that the proposed protocol effectively rewards the peers which share large number of authentic files in the network, which in turn helps in making the searching protocol efficient.

Fig. 10 shows variation of the value of HM for both the types of peers. Although, the value of HM for the good peers reaches a steady state as the topology matures, for the malicious peers, the value of HM fluctuates quite appreciably. The HM for the malicious peers sometimes attains higher values than that of the good peers. Since the queries forwarded by the malicious peers are blocked, HM for these peers are sometimes higher than those of the honest peers. The *hit* here does not mean authentic hit. The authentic hit of the good peers is higher than that of the malicious peers as these peers have higher AR values.

Fig. 11 shows the variation of the RIC for each type of peers under threat model *A*. It may be observed that the RIC for the good peers increases to 2.4 (constrained by the parameter *edge limit*), whereas for the malicious peers, the RIC does not increase beyond 1.2. With the increase in the percentage of malicious peers, the saturation rate slows down albeit the final value remains

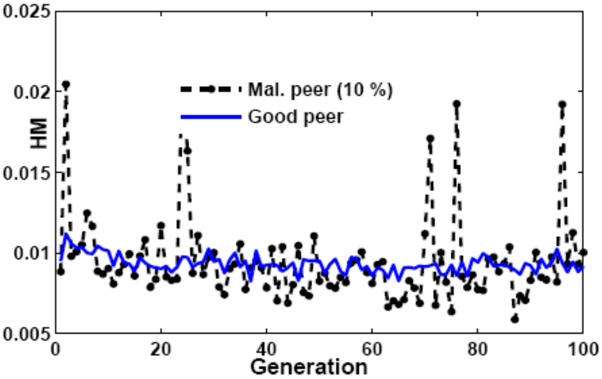


Fig. 10 HM for the malicious and the honest peers in the network. Percentage of malicious peers in the network is 10.

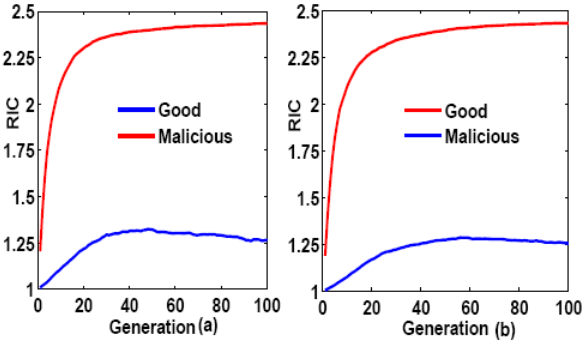


Fig. 11 RIC for various percentages of malicious peers under *threat model A*. In (a) 20% and in (b) 40% peers in the network are malicious.

the same. This shows that the proposed protocol provides better connectivity to the peers which share large number of authentic files. At the same time, the malicious peers are blocked gradually and their community edges are deleted.

Fig. 12 shows the variation of RIC under threat model *B*. Since in this model, a malicious peer provides fake files after it has achieved a high connectivity and then stops acting maliciously when it has lost sufficient number of community edges, fluctuation in the RIC persists throughout the simulation period.

Fig. 13 presents how the *closeness centrality* (CCen) of the good and the malicious peers varies in the community topology. In computation of CCen, only the community edges have been considered. It may be observed that the steady state value of the CCen for the good peers is around 0.12. However, for the malicious peers, the CCen value is found to lie in between 0.03 to 0.07. This demonstrates that the malicious peers are driven to the fringe of

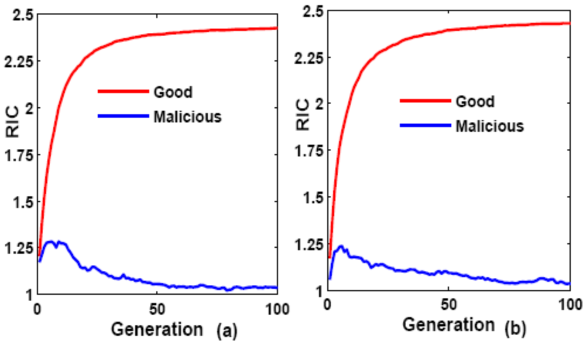


Fig. 12 RIC for various percentages of malicious peers under *threat model B*. In (a) 20% and in (b) 40% peers in the network are malicious.

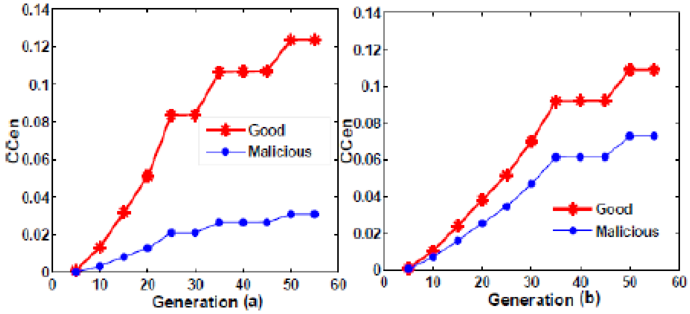


Fig. 13 Closeness centrality for various percentages of malicious peers in the network. In (a) 20% and (b) 40% nodes are malicious.

the network, while the good peers are allowed to form communities among them.

Higher values of CCen also indicate that the good peers have smaller average shortest path length between them. In the simulation, the diameter of the initial network is taken as 5. At the end of a simulation run, if there is no path between a pair of peers using the community edges, then the length of the shortest path between that pair is assumed to be arbitrarily long, say 15 (used in Fig. 14). As shown in Fig. 14, the *average shortest path distance* (ASPD) decreases from the initial value of 15 for both the honest and the malicious nodes. However, the rate and the extent of decrease for the good peers are much higher due to the formation of the semantic communities around them. For the malicious peers, after an initial fall, the value of ASPD increases consistently and finally almost reaches the maximum value of 15. On the other hand, the average value of ASPD for good peers is observed to be around 6. Since the good peers are connected with shorter paths, the query propagations and their responses will also be faster among these peers.

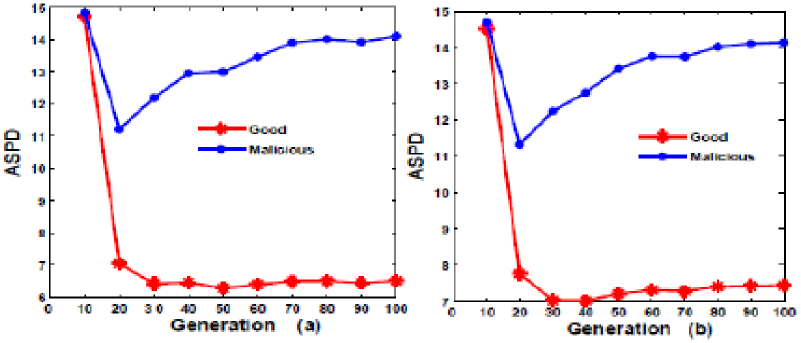


Fig. 14 Avg. shortest path distance vs. generations of search at the step of ten for various percentages of malicious peers. In (a) 30% and in (b) 40% nodes are malicious.

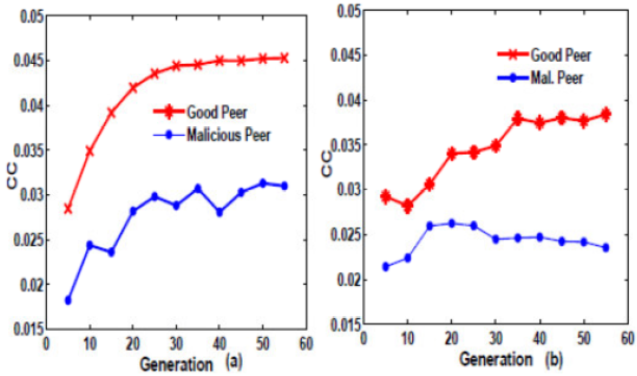


Fig. 15 Clustering coefficient for different percentages of malicious peers in the network. In (a) 20% and in (b) 40% of the peers are malicious.

Fig. 15 shows *clustering coefficient* (CC) for each type of peers. Since the community edges are added based on the download history and the peers having good reputation gain more community edges, the CC is high for the honest peers. This leads to the formation of triangles in the peer communities. To counter this phenomenon, the search strategy adapts itself from the BFS to the DFS to minimize redundant message flows in the network. Since the edges are added based on the download history and similarity of interest, the communities of the peers are formed which are connected to other community by hub of peers having interest in multiple content categories. This leads to lower ASPD for the good peers.

Fig. 16 depicts the size of the *largest connected component* (LCC) for each of the 32 content categories. It may be observed that the average size of the LCC for all content categories remains constant even if the percentage of the malicious peers in the network increases. This clearly shows that the community formation among the good peers is not adversely affected by the presence of the malicious peers.

The average value of the LCC in the proposed protocol is further compared with that of an equivalent graph for various percentages of the malicious peers and the results are presented in Fig. 17. It may be observed in Fig. 17 that the value of the LCC in the proposed protocol remains almost constant irrespective of the percentage of the malicious peers in the network. However, the average LCC in the equivalent network (without the proposed protocol) falls sharply with the increase in the percentage of the malicious peers. This clearly shows that the proposed protocol is effective in forming the semantic communities for all types of file categories even in the presence of high percentage of the malicious peers.

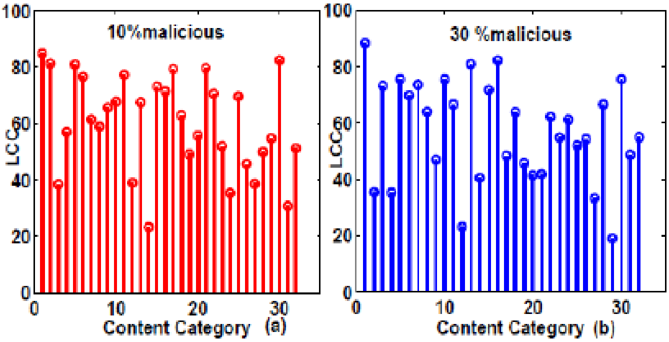


Fig. 16 Largest connected components (LCC) for different content categories

Fig. 18 shows that as the topology of the network matures, the steady state value of the *trust query propagation overhead* (TQPO) becomes quite low. The value of TQPO is less than 10 when 10% of the peers in the network are malicious. Even when the network has 40% of its peers malicious, TQPO gradually decreases and reaches a value of 20 in 100 generations. Hence, the trust propagation module has little impact on the system overhead, since the trust information is efficiently distributed in the trust-aware overlay topology.

The overhead due to the topology adaptation in the proposed protocol is also investigated. As mentioned in Section 4, the overhead due to the topology adaptation is measured by the metric TAO, which is defined as the number of community edges added or deleted in a generation. Fig. 19 shows the variation of TAO for different percentages of the malicious peers. It is observed that the value of TAO starts falling from an initial high value and oscillates with small amplitudes. This is due to the fact that initially the edge capacities of the peers are not saturated and they acquire the community edges

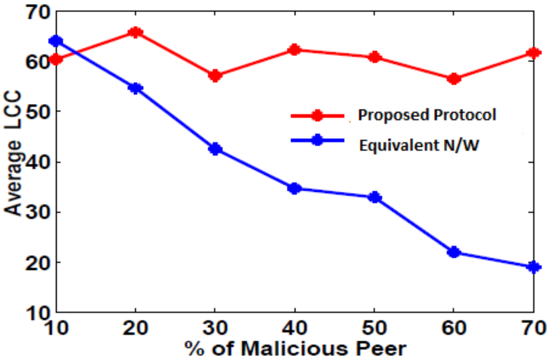


Fig. 17 Avg. LCC for different percentages of malicious peers in the network with the proposed protocol and without the protocol in an equivalent network

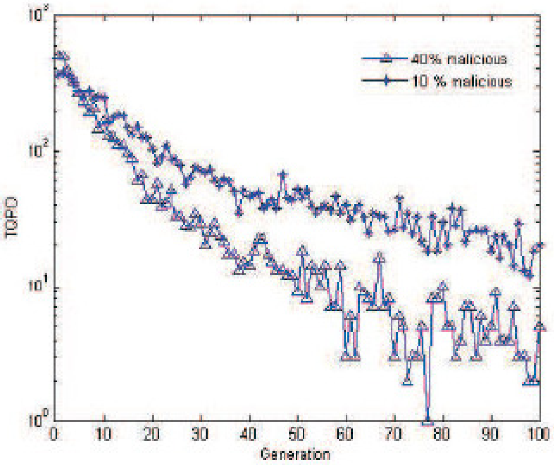


Fig. 18 Overhead of trust query propagation for 10% and 20% malicious peers in the network

rapidly. As the protocol executes further, the good peers acquire relatively stable neighborhood resulting in a sharp decrease in the value of TAO. In the subsequent generations, the value of TAO fluctuates slightly since the good peers delete the existing edges with the malicious peers as soon as the malicious peers are detected, and the new community edges with the fellow good peers are added. With the increase in percentage of the malicious peers, the fluctuation in the values of TAO also increases as more number of peers get added and deleted in the network. However, in all cases, the value of TAO falls sharply and attains a very low value once the community topology of

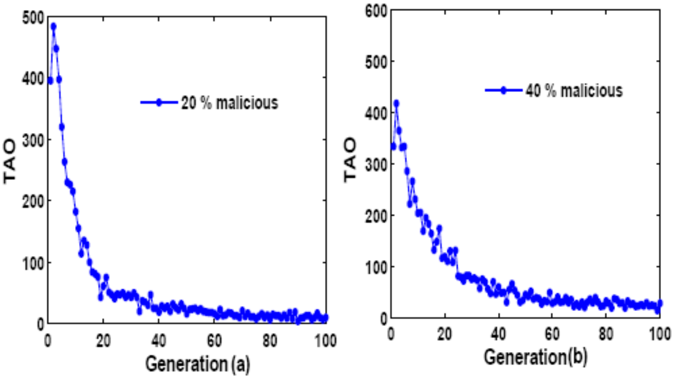


Fig. 19 Overhead due to topology adaptation under the presence of various percentages of malicious peers. In (a) 20% and in (b) 40% of the peers in the network are malicious.

Table 3 Avg. LCC values for different percentages of node churning

% of Node Churn	Avg. LCC
10	58
20	46
30	40
40	35
50	28

the peers becomes stable. This shows that the proposed protocol introduces a very small overhead in computation for topology adaptation process.

We have also studied how effectively the proposed protocol distinguishes the free riders from the malicious peers. As mentioned in Section 1, the free riders are those peers who do not share any resources with the other peers but they enjoy the resources of the other peers in the network. In the simulation, we have modeled the free riders as those peers who share up to 10 files in the content distribution model. The percentage of the free riders in the network is taken as 40. Since the free riders do not share any files, these peers will not be able to form any semantic communities. Accordingly, the RIC for these peers should be as low as those of the malicious peers. On the other hand, since unlike the malicious peers, the free riders do not distribute any spurious files, their presence does not cause much adverse impact on the network services. Hence, these peers are not penalized as much as the malicious peers. Accordingly, the AR values for the free riders should be higher than those of the malicious peers. The results presented in Fig. 20 validate this hypothesis.

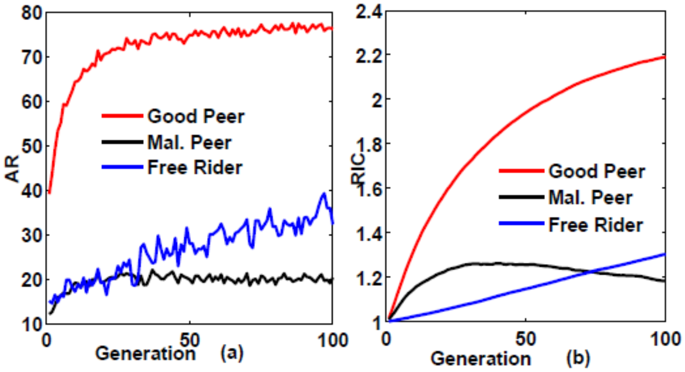


Fig. 20 AR and RIC values for the good peers, the free riders and the malicious peers for various percentages of malicious peers in the network. The percentage of free riders is taken as 40.

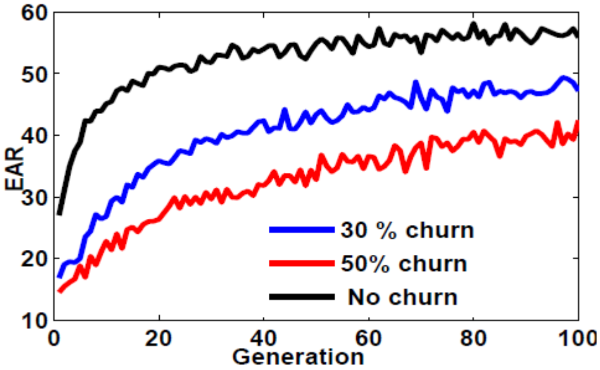
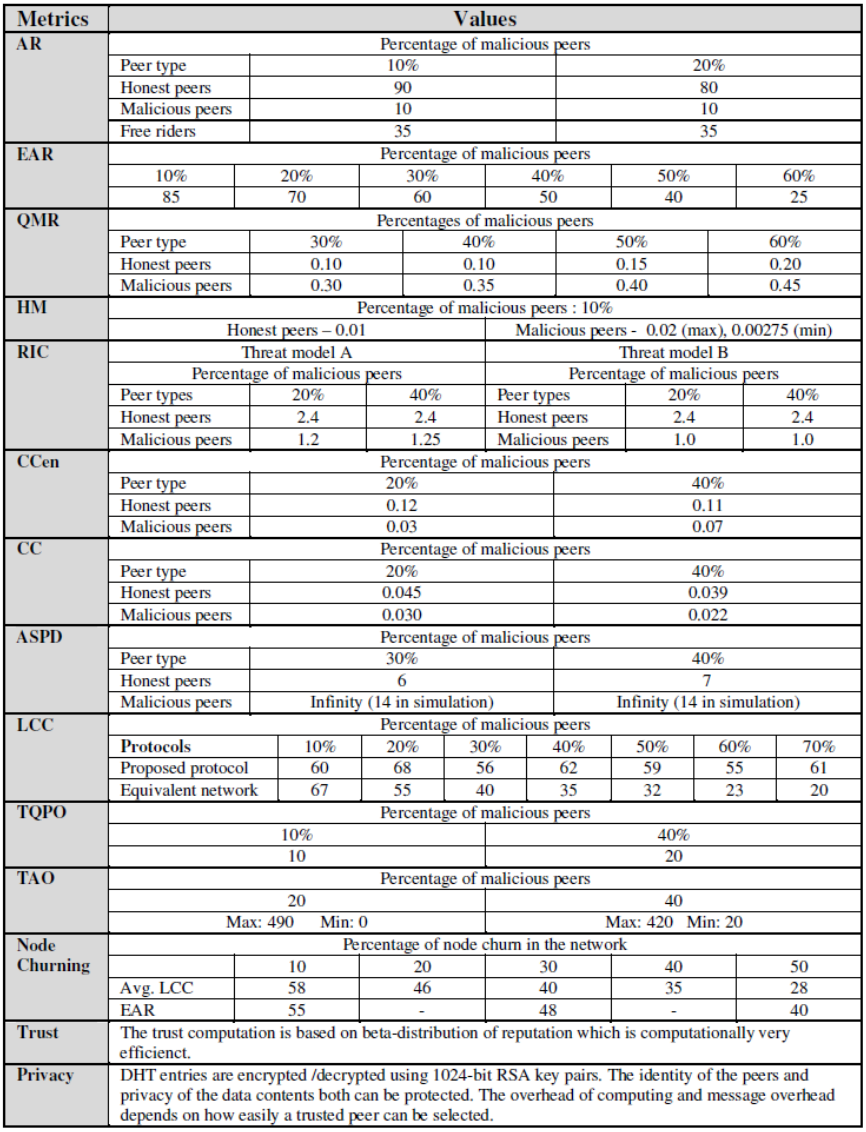


Fig. 21 Effect of node churning on EAR for various percentages of node churning with the proposed protocol in operation

Finally, we have studied the effect of *node churn* in the performance of the proposed protocol. Since, node churn is a natural phenomenon in a P2P network, it is essential that the search protocols should be efficient in the event of an occurrence of a large degree of node churn. Node churn causes disruption in the semantic communities of the peers. Hence, it leads to increase in the value of the QMR and decrease in the value of the EAR. Table 3 presents the average size of the LCC for various levels of node churn in the network, when the proposed protocol is under operation. Fig. 21 shows the effect of node churn on the EAR for various percentages of node churning. It may be observed that the performance of the proposed protocol degrades gracefully with the increase in the percentage of nodes being churned.

The performance of the proposed search protocol is summarized in Fig. 22.



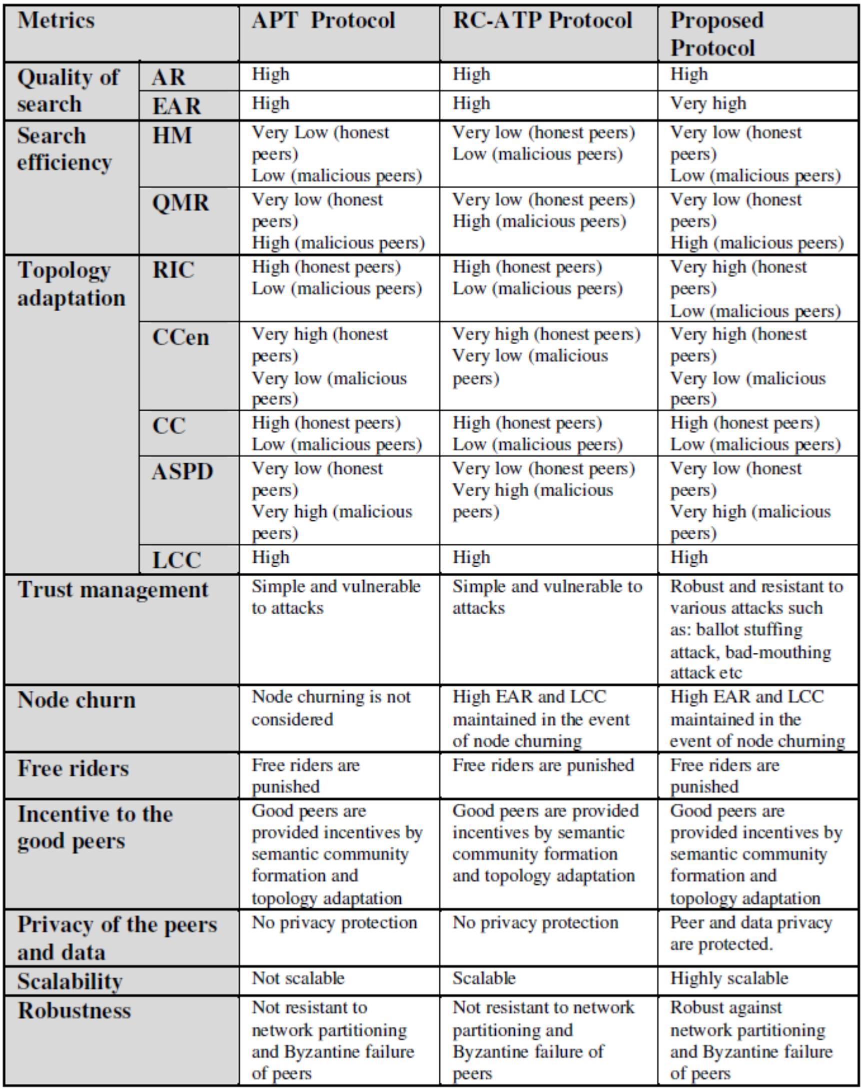
Metrics	Values						
AR	Percentage of malicious peers						
	Peer type	10%	20%				
	Honest peers	90	80				
	Malicious peers	10	10				
	Free riders	35	35				
EAR	Percentage of malicious peers						
	10%	20%	30%	40%	50%	60%	
	85	70	60	50	40	25	
QMR	Percentages of malicious peers						
	Peer type	30%	40%	50%	60%		
	Honest peers	0.10	0.10	0.15	0.20		
	Malicious peers	0.30	0.35	0.40	0.45		
HM	Percentage of malicious peers : 10%						
	Honest peers – 0.01	Malicious peers - 0.02 (max), 0.00275 (min)					
RIC	Threat model A			Threat model B			
	Percentage of malicious peers			Percentage of malicious peers			
	Peer types	20%	40%	Peer types	20%	40%	
	Honest peers	2.4	2.4	Honest peers	2.4	2.4	
	Malicious peers	1.2	1.25	Malicious peers	1.0	1.0	
CCen	Percentage of malicious peers						
	Peer type	20%	40%				
	Honest peers	0.12	0.11				
	Malicious peers	0.03	0.07				
CC	Percentage of malicious peers						
	Peer type	20%	40%				
	Honest peers	0.045	0.039				
	Malicious peers	0.030	0.022				
ASPD	Percentage of malicious peers						
	Peer type	30%	40%				
	Honest peers	6	7				
	Malicious peers	Infinity (14 in simulation)	Infinity (14 in simulation)				
LCC	Percentage of malicious peers						
	Protocols	10%	20%	30%	40%	50%	60%	70%
	Proposed protocol	60	68	56	62	59	55	61
	Equivalent network	67	55	40	35	32	23	20
TQPO	Percentage of malicious peers						
	10%	40%					
	10	20					
TAO	Percentage of malicious peers						
	20	40					
	Max: 490 Min: 0	Max: 420 Min: 20					
Node Churning	Percentage of node churn in the network						
		10	20	30	40	50	
	Avg. LCC	58	46	40	35	28	
	EAR	55	-	48	-	40	
Trust	The trust computation is based on beta-distribution of reputation which is computationally very efficienct.						
Privacy	DHT entries are encrypted /decrypted using 1024-bit RSA key pairs. The identity of the peers and privacy of the data contents both can be protected. The overhead of computing and message overhead depends on how easily a trusted peer can be selected.						

Fig. 22 Summary of the performance metrics of the proposed search protocol

Comparisons with Existing Protocols: In the following, we provide a brief comparative analysis of the proposed protocol with two similar protocols existing in the literature. In [34], a method named *eigen trust* has been proposed to minimize the impact of the malicious peers on the performance of a P2P system. In this scheme, the global trust value for each peer is computed by calculating the left principal eigen vector of a matrix of normalized local trust values. Since the trust and reputation computations are robust, the mechanism is able to sustain a high value of the AR (i.e. the fraction of authentic file downloads) for the good peers even when the percentage of the malicious peers is as high as 80. In contrast, the proposed protocol in this chapter can support a high value of AR for the good nodes as long as the percentage of the malicious peers in the network does not exceed 60. However, the scheme based on eigen trust is computationally intensive, and it is susceptible to produce unreliable results in the event of any Byzantine failures of some of the peers. On the other hand, the proposed protocol in this chapter has a light-weight trust management module that is robust yet efficient in identifying the free riders and Byzantine peers while improving on the QoS of searching.

In the APT protocol [16], as the topology stabilizes, all the paths from the good peers to the malicious peers are blocked, and the characteristic path lengths of these two types (good and malicious) of peers are distinctly different - while the good peers have shorter path lengths between them, the malicious peers are driven to the fringe of the network. However, in the proposed protocol in this chapter, the good peers and the malicious peers still remain connected through the connectivity edges since these edges are not deleted during the protocol operation. The presence of the connectivity edges prevents any possibility of network partitioning, which makes the protocol more robust and fault-tolerant. Moreover, the scalability of the proposed protocol is higher than that of the APT protocol, since it uses a light-weight trust engine. More importantly, the APT protocol does not have any mechanism to protect the privacy of the peers. The proposed protocol provides a very robust and reliable mechanism for protecting the privacy of the peers and their data. This makes it more suitable for deployment in the real-world P2P networks.

A comparative analysis of three protocols - the APT protocol [16], the RC-ATP protocol [55], and the proposed protocol in this chapter- is presented in Fig. 23. It can be observed that the proposed protocol outperforms the other two protocols in terms of its higher scalability, robustness against network partitioning, and its ability protect privacy of the peers and the messages communicated in the network.



Metrics		APT Protocol	RC-ATP Protocol	Proposed Protocol
Quality of search	AR	High	High	High
	EAR	High	High	Very high
Search efficiency	HM	Very Low (honest peers) Low (malicious peers)	Very low (honest peers) Low (malicious peers)	Very low (honest peers) Low (malicious peers)
	QMR	Very low (honest peers) High (malicious peers)	Very low (honest peers) High (malicious peers)	Very low (honest peers) High (malicious peers)
Topology adaptation	RIC	High (honest peers) Low (malicious peers)	High (honest peers) Low (malicious peers)	Very high (honest peers) Low (malicious peers)
	CCen	Very high (honest peers) Very low (malicious peers)	Very high (honest peers) Very low (malicious peers)	Very high (honest peers) Very low (malicious peers)
	CC	High (honest peers) Low (malicious peers)	High (honest peers) Low (malicious peers)	High (honest peers) Low (malicious peers)
	ASPD	Very low (honest peers) Very high (malicious peers)	Very low (honest peers) Very high (malicious peers)	Very low (honest peers) Very high (malicious peers)
	LCC	High	High	High
Trust management		Simple and vulnerable to attacks	Simple and vulnerable to attacks	Robust and resistant to various attacks such as: ballot stuffing attack, bad-mouthing attack etc
Node churn		Node churning is not considered	High EAR and LCC maintained in the event of node churning	High EAR and LCC maintained in the event of node churning
Free riders		Free riders are punished	Free riders are punished	Free riders are punished
Incentive to the good peers		Good peers are provided incentives by semantic community formation and topology adaptation	Good peers are provided incentives by semantic community formation and topology adaptation	Good peers are provided incentives by semantic community formation and topology adaptation
Privacy of the peers and data		No privacy protection	No privacy protection	Peer and data privacy are protected.
Scalability		Not scalable	Scalable	Highly scalable
Robustness		Not resistant to network partitioning and Byzantine failure of peers	Not resistant to network partitioning and Byzantine failure of peers	Robust against network partitioning and Byzantine failure of peers

Fig. 23 A comparative analysis of three protocols - APT, RC-ATP and the proposed protocol

6 Conclusion

In many IoT applications, resource discovery protocols are required which need to perform efficiently in a distributed and large-scale environment. An efficient and secure search protocol for unstructured P2P networks will be an ideal candidate for this purpose. Hence, the P2P architectures and their protocols are finding increasing relevance and adoption in IoT middleware design. In this chapter, we have presented a search protocol for unstructured P2P networks that solves several problems e.g., inauthentic downloads, poor search scalability, combating free riders, and protecting the user and the data privacy. The protocol exploits the topology adaptation done by the peers and uses a robust trust management mechanism to isolate the malicious peers while providing topologically advantageous positions to the good peers. Due to the topology adaptation, the good peers are able to form semantic communities which enables them get faster and authentic responses to their queries. On the other hand, the malicious peers are driven to the fringes of the network so that the queries from these peers have longer paths to travel to receive responses. In some situations, the queries from the malicious peers are blocked so that these peers do not receive any response to their queries at all. A large number of metrics are defined for evaluating the performance of the proposed protocol, and the protocol is simulated in a power-law network. The simulation results have demonstrated that the protocol is robust even in presence of a large percentage of malicious peers in the network. A detailed comparative analysis of the performance of the protocol is made with two existing similar protocols so that the advantages of the proposed protocol can be clearly understood. As a future plan of work, we intend to carry out an analysis of the message overhead of the privacy module under different network topologies and for different selection strategies of the trusted peers.

References

1. Abdul-Rahman, A., Hailes, S.: A Distributed Trust Model. In: Proceedings of the Workshop on New Security Paradigms (NPW 1997), Langdale, Cumbria, United Kingdom, pp. 48–60 (1997)
2. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM 2001), Atlanta, Georgia, USA, pp. 310–317 (2001)
3. Aberer, K.: P-Grid: A Self-Organizing Access Structure for P2P Information Systems. In: Batini, C., Giunchiglia, F., Giorgini, P., Mecella, M. (eds.) CoopIS 2001. LNCS, vol. 2172, pp. 179–194. Springer, Heidelberg (2001)
4. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in Poer Law Networks. Physics Review E64, 46135–46143 (2001)
5. Atzori, L., Lera, A., Morabito, G.: The Internet of Things: A Survey. Computer Networks 54(15), 2787–2805 (2010)
6. Balfe, S., Lakhani, A.D., Paterson, K.G.: Trusted Computing: Providing Security for Peer-to-Peer Networks. In: Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005), Konstanz, Germany, pp. 117–124 (2005)
7. Bandyopadhyay, D., Sen, J.: Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Personal Communications, Special Issue on Distributed and Secure Cloud Clustering (DISC) 58(1), 49–69 (2011)
8. Barabasi, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)

9. Bloom, B.: Space-Time Trade-Offs in Hash Coding with Allowable Errors. Communications of the ACM 13(7), 422–426 (1970)
10. Buchegger, S., Boudec, J.Y.L.: The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks. In: Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2003), Sophia-Antipolis, France, pp. 131–140 (2003)
11. Buchegger, S., Boudec, J.Y.L.: Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad-hoc Networks. EPFL Technical Report No: IC/2003/50 (2003)
12. Cha, B.R., Kim, J.G.: Handling Fake Multimedia Contents Threat with Collective Intelligence in P2P File Sharing Environments. In: Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2010), Fukuoka, Japan, pp. 258–263 (2010)
13. Chaum, D.: The Dining Cryptographers Problem: Uncontrolled Sender and Recipient Untraceability. Journal of Cryptology 1(1), 65–75 (1998)
14. Clarke, I., Miller, S., Hong, T., Sandberg, O., Wiley, B.: Protecting Free Expression Online with FreeNet. IEEE Internet Computing 6(1), 40–49 (2002)
15. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: Federrath, H. (ed.) Anonymity 2000. LNCS, vol. 2009, p. 46–66. Springer, Heidelberg (2001)
16. Condie, T., Kamvar, S.D., Garcia-Molina, H.: Adaptive Peer-to-Peer Topologies. In: Proceedings of the 4th International Conference on Peer-to-Peer Computing (P2P 2004), Zurich, Switzerland, pp. 53–62 (2004)
17. Crespo, A., Garcia-Molina, H.: Semantic Overlay Networks for P2P Systems. In: Moro, G., Bergamaschi, S., Aberer, K. (eds.) AP2PC 2004. LNCS (LNAI), vol. 3601, pp. 1–13. Springer, Heidelberg (2005)
18. Dellarocas, C.: Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior. In: Proceedings of the 2nd ACM Conference on Electronic Commerce (EC 2000), Minneapolis, MN, USA, pp. 150–157 (2000)
19. Damiani, E., di Vimecati, D.C., Paraboschi, S., Samarati, P., Violante, F.: Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washington DC, USA, pp. 207–216 (2002)
20. de Mello, E.R., van Moorsel, A., da Silva Fraga, J.: Evaluation of P2P Search Algorithms for Discovering Trust Paths. In: Wolter, K. (ed.) EPEW 2007. LNCS, vol. 4748, pp. 112–124. Springer, Heidelberg (2007)
21. Desmed, Y.G.: Threshold Cryptography. European Transactions on Telecommunications 5(4), 449–457 (1994)
22. Dingledine, R., Freedman, M.J., Molnar, D.: Accountability Measures for Peer-to-Peer Systems. In: Peer-to-Peer: Harnessing the Power of Disruptive Technologies, ch. 16. O'Reilly and Associates (2000)
23. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
24. Ganeriwal, S., Srivastava, M.B.: Reputation-Based Framework for High Integrity Sensor Networks. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2004), Washington DC, USA, pp. 66–77 (2004)

25. Ganeriwal, S., Balzano, L.K., Srivastava, M.B.: Reputation-Based Framework for High Integrity Sensor Networks. ACM Transactions on Sensor Networks (TOSN) 4(3), Article No. 15 (2008)
26. Gkantsidis, C., Mihail, M., Saberi, A.: Hybrid Search Schemes for Unstructured Peer-to-Peer Networks. In: IEEE INFOCOM (2005)
27. Goel, S., Robson, M., Pole, M., Sirer, E.: Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Cornell University, CIS Technical Report TR2003-1890 (2003)
28. Goldschlag, D., Reed, M., Syverson, P.: Onion Routing for Anonymous and Private Internet Connections. Communications of the ACM 42(2), 39–41 (1999)
29. Guo, L., Yang, S., Guo, L., Shen, K., Lu, W.: Trust-Aware Adaptive P2P Overlay Topology Based on Super-Peer-Partition. In: Proceedings of the 6th International Conference on Grid and Cooperative Computing (GCC 2007), Urumchi, Xinjiang, China, pp. 117–124 (2007)
30. Hsiao, H.C., Liao, H., Huang, C.C.: Resolving the Topology Mismatch Problem in Unstructured Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 20(11), 1668–1681 (2009)
31. Huang-Fu, C.C., Lin, Y.B., Rao, H.: IP2P: A Peer-to-Peer System for Mobile Devices. IEEE Wireless Communications 16(2), 30–36 (2009)
32. Joung, Y.J., Lin, Z.W.: On the Self-Organization of a Hybrid Peer-to-Peer System. Journal of Network and Computer Applications 33(2), 183–202 (2010)
33. Jsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
34. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigen Trust Algorithm for Reputation Management in P2P Networks. In: Proceedings of the 12th International Conference on World Wide Web (WWW 2003), Budapest, Hungary, pp. 640–651 (2003)
35. Leighotn, F.T., Rao, S.: An Approximate Max-Flow Min-Cut Theorem for Uniform Multicommodity Flow Problem with Applications to Approximate Algorithms. In: Proceedings of the 29th IEEE Symposium on Foundations of Computer Science (FOCS 1988), pp. 422–431 (1988)
36. Li, X., Wang, J.: A Global Trust Model of P2P Network Based on Distance-Weighted Recommendation. In: Proceedings of IEEE International Conference of Networking, Architecture, and Storage (NAS 2009), Zhang Jia Jie, Hunan, China, pp. 281–284 (2009)
37. Li, Z., Xie, G., Li, Z.: Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems. IEEE Transactions on Parallel and Distributed Systems 19(12), 1695–1708 (2008)
38. Lin, T., Lin, P., Wang, H., Chen, C.: Dynamic Search Algorithm in Unstructured Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 20(5), 654–666 (2009)
39. Lu, Y., Wang, W., Xu, D., Bhargava, B.: Trust-Based Privacy Preservation for Peer-to-Peer Data Sharing. IEEE Transaction on Systems Man and Cybernetics (Special issues based on best papers in Secure Knowledge Management Conference) 36(3), 498–502 (2006)
40. Martinez-Yelmo, I., Bikfalvi, A., Cuevas, R., Guerrero, C., Garcia, J.: H-P2PSIP: Interconnection of P2PSIP Domains for Global Multimedia Services Based on a Hierarchical DHT Overlay Network. Computer Networks 53(4), 556–568 (2009)

41. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content Addressable Network. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2001), San Diego, California, USA, pp. 161–172 (2001)
42. Reed, M., Syverson, P., Goldschlag, D.: Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications. Special Issue on Copyright and Privacy Protection 16(4), 482–494 (1998)
43. Reiter, M.K., Rubin, A.D.: Crowds: Anonymity for Web Transactions. ACM Transactions on Information and Systems Security 1(1), 66–92 (1998)
44. Risson, J., Moors, T.: Survey of Research Towards Robust Peer-to-Peer Networks. Computer Networks 50(7), 3485–3521 (2006)
45. Santucci, G.: From Internet of Data to Internet of Things. In: Proceedings of the 4th International Conference on Future of Internet Technology, Seoul, Korea (2009)
46. Scarlata, V., Levine, B., Shields, C.: Responder Anonymity and Anonymous Peer-to-Peer File Sharing. In: Proceedings of IEEE International Conference on Network Protocols (ICNP 2001), Riverside, CA, USA, p. 272 (2001)
47. Schafer, J., Malinks, K., Hanacek, P.: Peer-to-Peer Networks Security. In: Proceedings of the 3rd International Conference on Internet Monitoring and Protection (ICIMP 2008), Bucharest, Romania, pp. 74–79 (2008)
48. Schlosser, M.T., Condie, T.E., Kamvar, S.D., Kamvar, A.D.: Simulating a P2P File-Sharing Network. In: Proceedings of the 1st Workshop on Semantics in P2P and Grid Computing, Budapest, Hungary (2002)
49. Sen, J.: A Secure and Efficient Searching Scheme for Trusted Nodes in a Peer-to-Peer Network. In: Herrero, Á., Corchado, E. (eds.) CISIS 2011. LNCS, vol. 6694, pp. 100–108. Springer, Heidelberg (2011)
50. Sen, J.: Secure and User-Privacy Preserving Searching in Peer-to-Peer Networks. International Journal of Communication Networks and Information Security (IJCNIS) 4(1), 29–40 (2012)
51. Shafer, G.: A Mathematical Theory of Evidence. Princeton University (1976)
52. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
53. Sherwood, R., Bhattacharjee, B., Srinivasan, A.: P5: A Protocol for Scalable Anonymous Communication. In: Proceedings of IEEE Symposium on Security and Privacy, Oakland, CA, pp. 53–65 (2002)
54. Sit, E., Morris, R.: Security Considerations for Peer-to-Peer Distributed Hash Tables. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 261–269. Springer, Heidelberg (2002)
55. Tain, H., Zou, S., Wang, W., Cheng, S.: Constructing Efficient Peer-to-Peer Overlay Topologies by Adaptive Connection Establishment. Computer Communication 29(17), 3567–3579 (2006)
56. Tang, X., Xu, J., Lee, W.C.: Analysis of TTL-Based Consistency in Unstructured Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 9(12), 1683–1694 (2008)
57. Tang, Y., Wang, H., Dou, W.: Trust Based Incentive in P2P Network. In: Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business, Beijing, China, pp. 302–305 (2004)
58. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2001)
59. Waldman, M., Rubin, A.D., Cranor, L.F.: Publius: A Robust, Tamper-Evident, Censorship-Resistant, Web Publishing System. In: Proceedings of the 9th USENIX Security Symposium, Denver, Colorado, USA, pp. 59–72 (2000)

60. Xiao, L., Liu, Y., Lionel, M.N.: Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment. IEEE Transaction on Computers 54(9), 1091–1103 (2005)
61. Xie, C., Chen, G., Vandenberg, A., Pan, Y.: Analysis of Hybrid P2P Overlay Network Topology. Computer Communications 31(2), 190–200 (2008)
62. Xiong, L., Liu, L.: A Reputation-Based Trust Model for Peer-to-Peer E-Commerce Communities. In: Proceedings of the 4th IEEE/ACM Conference on E-Commerce (CEC 2003), Newport Beach, California, USA, pp. 228–229 (2003)
63. Xioreng, L., Liu, L.: Peer-Trust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)
64. Xiao, L., Xu, Z., Zhang, X.: Low-Cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 14(9), 829–840 (2003)
65. Yang, M., Yang, Y.: An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing. IEEE Transactions on Computers 59(9), 1158–1171 (2010)
66. Zhang, R., Hu, Y.C.: Assisted Peer-to-Peer Search with Partial Indexing. IEEE Transactions on Parallel and Distributed Systems 18(8), 1146–1158 (2007)
67. Zhu, Z., Kalnis, P., Bakiras, S.: DCMP: A Distributed Cycle Minimization Protocol for Peer-to-Peer Networks. IEEE Transactions on Parallel and Distributed Systems 19(3), 363–377 (2008)
68. Zhuge, H., Chen, X., Sun, X.: Preferential Walk: Towards Efficient and Scalable Search in Unstructured Peer-to-Peer Networks. In: Proceedings of the 14th International Conference on World Wide Web (WWW 2005), Chiba, Japan, pp. 882–883 (2005)

The Tailored Fabric of Intelligent Environments

James Dooley, Hani Hagras, Vic Callaghan, and Martin Henson

Abstract. The traditional Internet of Things (IoT) vision states that passive, everyday objects are uniquely identified through some computer-readable means such as barcodes or RFID so that electronic systems can identify them. The identity is then used to retrieve a virtual representation for the object - a source of information that forms the basis for context awareness, decision making or action invocatoin. It was envisioned that every object in the world could be tagged and that the Internet could provide the network across which these *"things"* could be active (resolved, interacted, etc.). In this chapter, we describe how this vision converges with the vision for Intelligent Environments (IEs) as Ubiquitous Computing deployments that are endowed with an Ambient Intelligence. In particular we see the marriage of passive-objects from IoT and active-objects from IE as symbiotic if real-world deployment can ever be achieved - it is from these objects that the fabric of IEs will be woven.

1 Introduction

As the vision for an Internet of Things (IoT) becomes closer to reality, the number of objects that are deployed in the real world with a digital presence increases towards a massive scale. Familiar objects that already exist around us in the spaces we occupy will be given a digital-identity and possibly endowed with computational and communication capabilities. They, along with new and novel objects that include the virtual, will be interconnected and reflected by a digital presence. Collectively, these objects form part of the IoT – a massive distributed system that requires infrastructure to enable operation, discovery and management while simultaneously protecting scope, security and privacy. With such a rich and diverse landscape of information, there arises the necessity for standard middleware and novel Artificial Intelligence (AI) to be used in dealing with / operating such a body of knowledge that results.

James Dooley · Hani Hagras · Vic Callaghan · Martin Henson
University of Essex, Wivenhoe Park, Colchester, Essex, U.K. CO4 3SQ.
e-mail: {jpdool,hani,vic,hensm}@essex.ac.uk

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 321–344.
DOI: 10.1007/978-3-642-34952-2_13

This chapter introduces the concepts that outline how active and passive objects are interconnected and unified in the real world. Collectively they form the fabric from which Intelligent Environments (IEs) are constructed and provide a layered support for intelligent agents and software applications to operate atop. The formation of an agent population within an IE results in an emergent and collective Ambient Intelligence (AmI) that exists as a product of interaction, cooperation and even competition among intelligent agents. The model scales-up to form a view of the world as a set of interconnected IEs between which human users and their subservient agents may roam. The purpose of this effort is to facilitate a better quality of life and continuity-of-experience as perceived by human inhabitants through environmental adaptation.

Herein, we describe convergence between the IoT and IE research fields. Although the two are distinct in their vision, the real world will be deployed by a hybrid of both - this chapter identifies how the envisioned Large-Scale Intelligent Environments (LSIEs) infrastructure can support the digital identity that is given to passive objects by mixed technologies such as RFID and barcodes. Conversely, the IoT vision enhances the operation of IEs by enabling passive objects to exist along-side active objects (which have embedded computation and communication capabilities).

Sect. 2 introduces the more significant IoT and IE literature that has led us to the present state-of-the-art. This includes a description of the four *"living-labs"* that we have constructed on the University of Essex campus. Sect. 3 then presents a view of the world that scales from individual passive / active objects, through the IEs they occupy and up to clouds as virtual collections of IEs. Sect. 4 describes the significant requirement on enabling technology - such as middleware for the interconnection of entities (Sect. 4.1) and agents, applications and virtual appliances for intelligent operation (Sect. 4.2). The material discussed is then illustrated through the use of a case study in Sect. 5. The chapter is then summarised and concluded with some remarks on future work in Sect. 6.

2 State of the Art

In 1991, Mark Weisers seminal work [1] described a grand *"Ubiquitous Computing"* (UC) vision for our future in which computer technology becomes transparently embedded in the world around us. This takes a user-centric approach in which the cognitive load of technology on people is reduced by making technology recede into the background of our lives, beyond human perception. This is in contrast to the modern day model of people staring *"awkwardly"* at a desktop screen - interacting on the terms of technology. The emergence of mobile-computing can be seen as a stepping stone between the two where users are always connected through the device they carry with them. In the years since, UC has flourished and stimulated a great many works that span the entire spectrum of technology and society. The related fields of UC, IE, IoT, AmI and *Pervasive Computing* all offer variations around the same theme with focus on specific parts of the problem space. In the following two sub-sections, we focus on the more significant works in the IE and IoT areas.

The convergence of these two areas is object-centric and relies on the availability of common infrastructure. The real world will likely be the result of work that evolves from both IE and IoT fields and so the convergence of the two must be achieved. The problems of infrastructure and sensor availability that afflict the IoT vision are implicitly solved by the IE vision, whilst the IE need for a mix of passive and active objects is augmented by the IoT body of knowledge.

2.1 Internet of Things

Circa 1999, the IoT concept was suggested as a means to connect the internet to the physical world through the large-scale deployment of sensors. The intended purpose was to remove the dependance of humans by machines in acquiring information and to allow the information to be directly sampled from the real-world. With this proposition, the *"Auto-ID Centre"* was established. The purpose of the Auto-ID Centre was focused towards the investigation into Radio-Frequency IDentification (RFID) technology so that everyday objects could be given an Electronic Product Code (EPC) to aid supply-chain management [2] [3]. The work carried out under this Centre surpassed the standardisation of RFID and also investigated other associated problems, such as the specification of a common description language used to describe objects, processes and environments [4].

The *"Cooltown"* project explored the possibilities of linking every object in the world to a web-presence [5]. The work made use of several contact and non-contact sensing technologies and was motivated to link the physical and virtual worlds towards a mobile computing vision. Of note, this work examined other works in the field and classified the nature of links between a physical and virtual object by application [6] :

- **Physical Browsing:** The association of digital documents with physical objects - *users designate entities that interest them, and thereby obtain documents (pages) about them.*
- **Content Repositories:** The association of some digital content with physical objects - *so that users may transfer the content to one another or move it from place to place by passing the corresponding object around.*
- **Copy-and-Paste:** The temporary association of content with a physical *"clipboard"* object - *so that users can copy content from a source and paste it to a sink.*
- **Communication Points:** The association of communication medium with physical objects - *so that users who encounter the same object can communicate (for example using bulletin boards, email, voicemail, etc.).*
- **Physical Icons:** The association of actions with physical objects - *so that users can invoke actions such as turning a light on, by manipulating a physical object (in a similar fashion as selecting an icon on a desktop PC will invoke some action).*

The authors of [6] also describe *Physical-Icons* as inputs to computational functions where the physical object can be mapped to a virtual entity. We would however argue that this deserves its own classification:

- **Object Reflection:** The association of a virtual *entity* with a physical object - *so that a user can identify an object, for the system to manipulate in some way through its virtual counterpart (for example identifying a camera will then provide a user-interface for the virtual functions of the camera, such as "view photos")*. The identification of users can also be considered as part of this classification.

Various technologies have been used to achieve the tagging of objects so that they can be identified uniquely, reliably and quickly by electronic systems [7]: Visual Object-recognition [8] [9], barcodes [10] [11], 2D barcodes (such as QR codes) [12], Infrared beacons [13] and badges [14], Ultrasound [15] [16], RFID [17], Ubisense (an RF-based realtime location tracking technology) [18], Wi-Fi [19], etc.

The salient point of the original IoT work is to identify *"passive"* objects (objects which have no computational or communication capabilities beyond that required to identify themselves) and to then do something based on that identity. It is worth noting that the tagged objects are passive - they do not actively do anything themselves, but rather the system that identifies them will carry out some action based on that recognition.

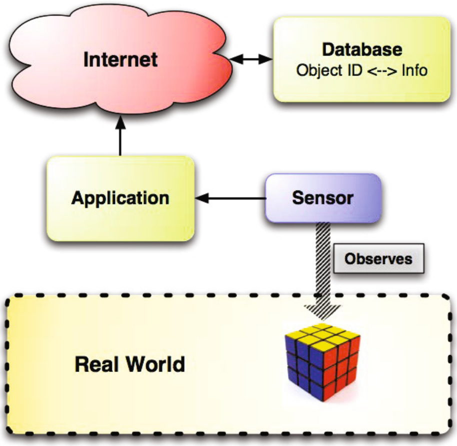


Fig. 1 The typical IoT architecture

Fig. 1 shows the typical architecture taken: *an application uses some sensing apparatus to observe an object in the real world. A database is then queried over the internet (or some other network) to resolve necessary information from the identity that has been gained from the real-world object. The application can then use this information to achieve its functional operations.*

As the work towards an IoT progresses with works such as [20], the vision evolves and converges with that of the IE field. In particular the inclusion of not just passive objects, but active ones as well is increasingly popular. This now causes confusion as to what the *"things"* in the IoT actually are [21] - *"are they sensors, are they devices, or are they passive objects?"*

2.2 Intelligent Environments

As a multi-disciplinary research area, there are a huge variety of topics into which IE researchers delve. Consequently, this results in diverse approaches being taken to construct *"living labs"* in which the research is conducted. For example, the Cisco *"Internet House"* was constructed on a full building scale, but its purpose was to showcase a home with always-on Internet connectivity and appliance automation (where the home and its appliances could be controlled over the internet). Similarly, the Philips *"HomeLab"* [22] was a fully functional apartment whose purpose was aimed more at user experience evaluation through the use of monitoring technologies (such as cameras and microphones). The greater extent of technology deployment in the MIT *"Placelab"* also took place in a dedicated apartment scale space and focused on the space construction, technology deployment and user experience. The Stanford *"iRoom"* [23] and National Institute for Science and Technology (NIST) *"smart space"* [24] have investigated the deployment of ubiquitous computing in the office / meeting room context. The Fraunhofer inHaus-Center run two labs called the *"SmartHome"* and *"SmartBuilding"* for research into many different areas of innovation including user interfaces [25], an area of research also investigated by the *"iRoom"* at LIMSI [26]. Facilities such as the Duke University *"Smart Home"* have been used primarily for student projects, while the more recent emergence of community-lead *"hackerspaces"* around the world have promoted public participation in technology-oriented projects. Being rich in interconnected computing devices, sensors and actuators, these *"technology rich"* environments are the precursors to the IE - lacking only a quality of intelligence that is achieved through the deployment of suitably endowed software, such as intelligent embedded-agents.

Recognising this disparity, researchers have deployed intelligence into numerous spaces. At the University of Colorado, the *"Adaptive Home"* used a centralised neural-network based controller that monitored approximately 75 sensors (light, temperature, sound, motion, door/window state, etc.) and then took appropriate action on related actuators in the home [27]. Over the lifetime of this lab, many experiments were conducted and results published regularly. Such a rich publication history also exists for the Georgia Institute of Technology *"Aware Home"* that explores a huge diversity of subject areas including sociological applications such as assisted-living and home-care [28]. The *"PEIS home"* at the Orebro University further extends the capability of environment manipulation that lies within control of software intelligence by deploying and integrating mobile robots into its infrastructure [29]]. Elegantly, some labs (such as the iRoom at the German University in Cairo [30] and the MavHome at Washington State University [31]) are used to experiment with populations of software agents that provide the ambient intelligence (this is especially interesting when considering emergent behaviour from populations of agents that compete or collaborate).

At the turn of the century, when technology became cheap, small and abundant, there was a renewed energy in the field of UC. Works such as [32] were stimulated and the *"disappearing computer"* [33] was being chased. Among the fray of projects that we spawned, the e-Gadgets (extrovert gadgets) project was started and focused on the creation of pro-active *"Intelligent Artefacts"* [34]. In support of this, the Intelligent Dormitory (also known as the *"iDorm"*) was constructed as a testbed that mimicked a single room student accommodation where individuals could stay for short periods of time (1-2 weeks). Within this seemingly normal place, heterogeneous technology was embedded and interconnected to form a grid computing deployment [35]. The iDorm identified and motivated continued work into the various challenges of UC, such as the Pervasive Home Environment Network (PHEN) project that continued to investigate the middleware and end-user interaction challenges [36].

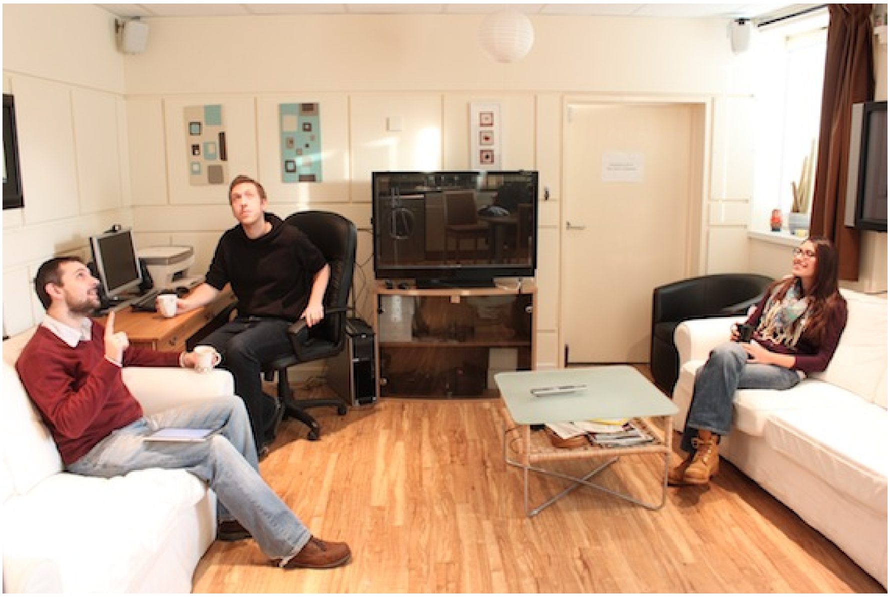


Fig. 2 The University of Essex iSpace (living-area)

A range of devices, technologies and networks were used and almost every aspect of the space could be monitored or controlled by the software agents that constantly executed and evolved. This has been previously and comprehensively described over years of publication, such as in [35] [37] [38]. In combination with desktop PCs and hand-held devices; motion, pressure, temperature and light sensors sampled the world, blinds could be opened / closed, lamps and lights could be dimmed / switched, doors could be unlocked, heaters / coolers could be controlled, etc. From this work, the *"iSpace"* (a fully functional apartment, shown in Fig. 2 discussed more in Sect. 5) and the *"iClassroom"* [39] were evolved.

The salient point to note about the IE field is that objects are considered *"active"* - they are envisioned to have embedded systems and communications capabilities within them and so are able to perform tasks themselves. The combination of infrastructure and a population of these active-objects results in a complex and dynamic distributed system - one that intelligent agents are envisioned to operate, thus giving spaces an AmI quality . The overall IE resource is itself intended to be adaptable to changes in context and user preference through software agents that not only sense the real world, but also act upon it through actuators (see Fig. 3).

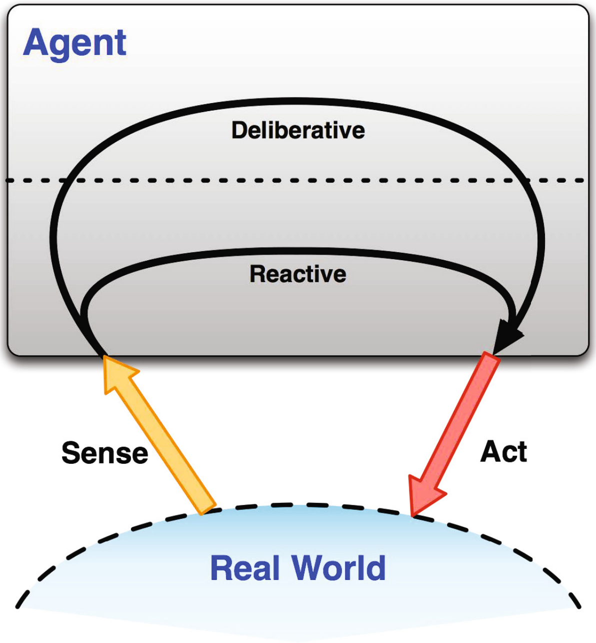


Fig. 3 The sense / act cycle that software agents conduct

3 The World View

Beyond the test-beds of research and proof-of-concept works; the realisation of the UC / IoT / IE visions in the real world rely on operation at large-scale. To date this is something that the IoT field has accomplished very well and that the IE field is only just beginning to venture into [40] [41]. From a top-down perspective, our future world can be seen as a set of geographically distributed IE *"Spaces"* - interconnected by the internet (Fig. 4). Users roam through the physical world entering and exiting these IEs in a transient nature - each user having a distinct *"role"* in each and carrying with them a digital profile that contains their data, agents, preferences, etc. [42] [43]. Users can then be considered to have a history of occupancy within a subset of the IE superset (i.e. through the life of a user, that user will have visited some of all the Spaces that exist within the world).

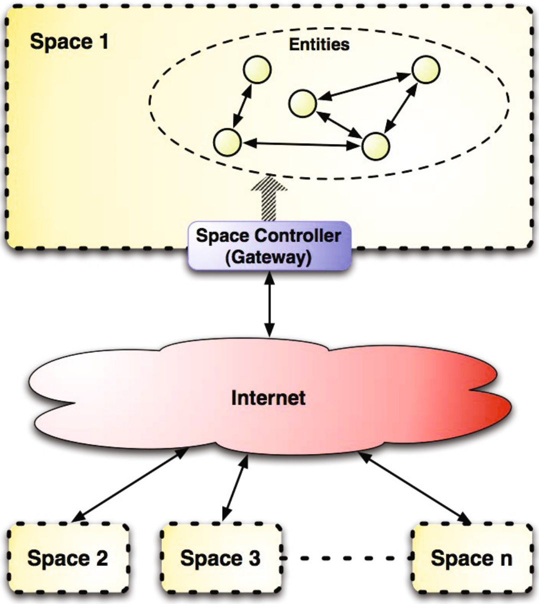


Fig. 4 An architecture for a *"world-of-spaces"*

Within this model, there are two distinct architectures that come together: the inter-Space and intra-Space . The inter-Space architecture is large-scale and formed by the interconnection of Spaces over a large network such as the global internet. The intra-Space architecture is concerned with how a space is composed from its constituent devices and entities. We use the concept of an abstract *"Entity"* [44] to describe the uniquely identifiable digital-presence for an object of some form (such as a sensor, actuator, file, process, user, place, etc.) regardless if it is real, virtual, logical or otherwise. Entities reside on physical devices and are grouped into sets that are published together by a *"Peer"* to a Space (Fig. 5).

In its simplest description; a Space (S) is a virtual machine (VM) that is distributed over a set of interconnected Peers that communicate through a network using secure middleware [45] . This virtual machine abstraction provides the conceptualisation that a space is centralised (with the associated advantages of management and security), even though it is indeed distributed (with the inherent properties of scalability and robustness). Several independent and isolated Spaces can exist across the same set of network-connected devices.

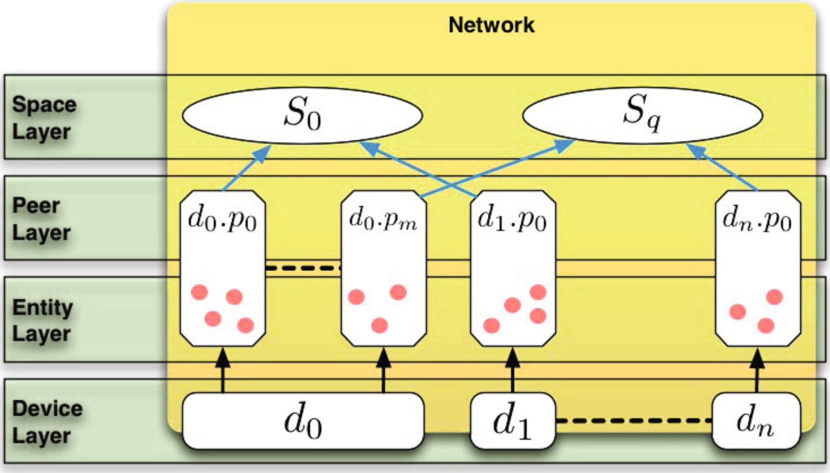


Fig. 5 Construction of a Space from its fabric (*"devices"*, *"peers"* and *"entities"*)

The Space Controller (SC; Fig. 4) represents the convergence of the inter-Space and intra-Space architectures. Itself an Entity; the SC acts as a gateway between locality and the wider large-scale. It also has the responsibility to manage the Space, its identity, members, etc. Through this gateway, Entities within a Space can be securely and safely accessed from outside. This model represents the convergence of IE and IoT towards a structured *"internet-of-entities"* .

4 Enabling Technology for IoT and IE convergence

A UC deployment is implicitly a distributed system - it relies on the interconnection of many computing devices and their software components across a network. Fig. 6 shows this and also shows the enabling technologies that are deployed across those devices. Of note, the middleware component (Sect. 4.1) must exist on every device that wishes to participate in the distributed system. This provides network transparency for the software that is deployed on top of the middleware - allowing higher level components to operate. Virtual Appliances can be formed, while Agents, Applications and entities can communicate to achieve behaviour and functionality within an IE (Sect. 4.2).

In the remainder of this section, these enabling technologies are presented.

4.1 Middleware

Middleware provides a common functionality to higher level software such as agents and applications while abstracting the underlying implementation. It is the enabling

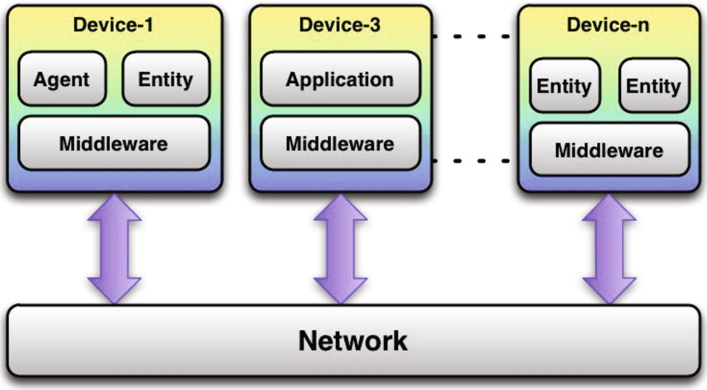


Fig. 6 A distributed system involving many devices, middleware, agents, applications and entities

technology that permits the processes on a single computing device to be rendered in a wider distributed computing environment - local resources can be exported and remote resources imported. As discussed in Sect. 3, the realisation of a converged IoT / IE reality depends on two kinds of architecture - the inter-Space and the intra-Space. While they must both support the same functionality (such as Entity discovery, interaction, eventing / subscription, etc.) they require slightly different approaches that are tailored to the conditions under which they must operate. The Space-Controller represents a convergence of these two approaches. At an inter-Space level, functionality is required to connect between spaces across the internet, while at the intra-Space level the emphasis is on the interconnection of Entities on a local network. The purpose of these architectures is to provide an end-to-end support for the interconnection of communicating entities that may reside in separate Spaces on a global scale.

Many approaches have been investigated for middleware that operates at the intra-Space level, [44] [46] but the core functionality that has been evolved here has not been scaled-up to an inter-Space level (although the proposition of this has been suggested [47], it is still an open and exciting area of investigation):

- **"Entity Discovery":** The ability to discover a previously unknown entity given some search parameters - this is a particularly difficult thing to achieve in a distributed system that is subject to any real entropy. On a large scale (such as searching web-pages on the internet for content) this is usually achieved by centralised *"Search Engines"*, while on the intra-Space level it is realistic to use distributed search requests through broadcast / multicast messaging.
- **"Entity Resolving":** The ability to resolve an entity identity to the current location of the entity so that further interaction may occur (by routing messages to it). On the WWW, a URL acts as both a page identity and location, but in the IE / IoT vision, entities may be mobile and move from location to location - identity and location should therefore be de-coupled.

- **"Interaction":** The ability to send messages to an entity and receive responses.
- **"Eventing":** The ability to subscribe to an entity so that it may send asynchronous messages back (this is in contrast to polling that is inefficient, particularly on a large scale).

Technically, the scope of approaches used to achieve functional middleware varies, but the two most common are Remote Procedure Call (RPC) and Message-Oriented Middleware (MOM). These are very distinct in their approach - the former treats remote objects like local ones and presents software with a proxy of some form upon which procedures can be invoked as if it were a local resource, while the latter achieves communications by routing messages between entities. More recently, the concept of leveraging web technologies (HTTP, SOAP, etc.) and applying them in a service oriented fashion has attracted much attention. Although most of the work in this area still focuses on the use of larger, more powerful desktop / server hardware, the performance limitations are plainly seen when attempting to apply the same techniques with embedded systems that are less capable of processing the comparatively large message sizes that are typically encoded in XML documents (despite some more recent work in the past few years towards overcoming this limitation [48] [49]).

It has become a popular practice to use these underlying technologies as simply a transport mechanism and expose an Object model to the higher levels through an API. This approach provides a more convenient / usable middleware (that can sometimes be swapped out for alternative middleware) for higher level software to utilise. As part of this, the middleware layer will typically also incorporate extra features to aid with reliability and quality-of-service such as automatic failure detection and selection of new candidates. In some cases, the higher level API actually dictates the underlying model and results in what has come to be known as Object-Oriented Middleware (OOM). The early Object Request Broker (ORB) approach [50] is quintessential of this kind of middleware and attempted to provide an implementation independent specification (through the Interface Description Language - IDL) from its inception. This approach is still popular and has resulted in many flavours of OOM, such as [51], while forming the basis for further investigations such as the emergence of Reflective middleware [52] that makes assurances regarding the fidelity between an Object and its remote representations on the client side.

It is well-understood that there is a necessity for entity identity that is unique across space and time in a distributed system. Active-objects pro-actively present their own identity to the Space in the form of an Entity , but passive-objects have no way to achieve this and rely on the infrastructure to carry out the correct actions following identification. Passive objects must therefore be resolved from their ID by using logic / knowledge that exists either within an application / agent, a Space, a user-profile or some other entity in the wider world. The effect of resolving a passive-object identity can vary depending on what that object ID links to and the context in which it is used (see Sect. 2.1: Physical-Browsing, Content-Repositories, Copy-and-Paste, Communication-Points, Physical-Icons, Object-Reflection). For example; an RFID tag that is linked to a user ID - when the tag is identified by an access-control agent (that identifies users at a door and controls the door lock), the agent will seek to establish if that user has permission to enter and either unlock the door or provide some feedback to the contrary. However, the same tag linked to the same user in a different context will have a different effect - for example if the tag is identified to a coffee table, then that table may undergo some adaptation such as display artwork / messages. Likewise, an RFID tag that is linked to a song / album and identified by the same table may display the artwork for that music and begin playback via a media control agent. This is further explored in Sect. 5.

4.2 Agents, Applications and Virtual Appliances

Across the large-scale of deployment that is the world of Spaces, there is a very large scope for *"things-that-do"* as consumers of existing information and producers of synthetic information. The purpose of software that falls under any of these categories is to achieve some functionality - i.e. to do something. The variation among them is due to *how* that something is done:

- **Applications:** These are pieces of software that are designed to achieve some specific function and are generally developed to operate in the same way as traditional distributed system software - interacting with distributed entities across a network. For example, a digital photo-frame that loads image entities across the network and shows them sequentially on a display. This kind of application may also expose some interface that allows other things-that-do to manipulate its behaviour (for example, to pause on the current photo or flick through to the next photo).
- **Agents:** These are somewhat more complex than standard applications; agents are embedded with some form of AI or computational intelligence and are characterised by being pro-active, that is they do not simply react to user control, but actively operate independant of it. Some, but not all, will also have a capability to learn from experience and self-adapt behaviour / structure. While they can operate independantly, there is also a huge scope for populations of interacting agents that cooperate and compete. Agents have been used extensively in the IE field, where they are given the ability to interact with the real world through sensors and acuators (see Fig. 3).
- **Virtual Appliances:** Virtual appliances can be constructed at runtime by linking together several component entities [53]. For example, a music-player can be constructed by linking a data source (for example an mp3 entity) to a decoder and then to one or more speakers. By assigning input and output *"ports"* to entities, simple graph-theory can be applied to construct a great number of appliances from the same set of component entities - it is the flow on information between them that achieves functionality. The appliances can be constructed, modified and deconstructed in real-time by simply linking / unlinking their IO ports. Hence, applications are recombinant [54].

All three kinds of things-that-do are portable across Spaces and can travel with a user from Space-to-Space. They do however all rely on middleware functionality to resolve component dependencies from those entities that are available at runtime (a process known as *"Runtime-Discovery"*). This concept can be extended to improve reliability / robustness to component failure by swapping out components for *"better"* ones as and when they are found.

While virtual appliances are essentially instantiated by the interconnection of entities, they are inherently bound to intra-Space deployment. Applications and agents can, however, reside *"in the cloud"* and peer into Spaces by interacting with entities that are accessible through the Space-Controller. This is especially useful considering that user-profiles will have some presence and dependance upon the cloud to facilitate the migration of digital assets from Space-to-Space.

A mixed population of things-that-do within an environment gives the user an experience that has a variable level of autonomy and transparency. This eases the cognitive load on the user by hiding away some decision-making and operation whilst making others overt. Filtering of user-direction makes the increasingly technological world more tractable without removing the sense of control that users need in order to be accepting of UC. In particular, as a user moves from Space-to-Space they experience a continuity of experience as the environment is adapted to the preferences of the user. This gives an impression that there is a collective and coordinated AmI at work on behalf of IE occupants.

5 Case Study: The Essex iSpace

The *iSpace* is a purpose-built, fully functional apartment that resides within the School of Computer Science and Electronic Engineering (CSEE) at the University of Essex, UK (see Fig. 2). Its layout consists of a main living area, kitchen, bedroom, study-room, bathroom and control room. A false ceiling and false walls provide additional space to hide technology such as sensors, actuators and computational devices from small embedded systems up-to full desktop PCs. As a UC deployment the iSpace is equipped with numerous sensors that sample the various phenomena of the real world, actuators that manipulate aspects of the environment, computational devices that run software and a firewall protected network that interconnects the entire resource. The architecture of this deployment is shown in Fig. 7, where the Ethernet / WiFi backbone can easily be seen as the convergence of many devices, some of which act as gateways into specific technologies. UPnP is deployed as a middleware that homogenises the heterogeneous and distributed UC resources, thus providing a consistent and accessible view of the network for software agents and applications. At the time of writing (March 2012), there are over 100 UPnP devices deployed within the iSpace network, each representing an entity of some form (logical, virtual, real, etc.). The availability of dynamic-discovery, event-subscription and action-invocation within this *living-lab* allows loosely-coupled agents and applications to interact directly with every entity on the network - UPnP device / service types define common interfaces; sensors can produce asynchronous events and actions can be invoked to achieve some function (such as turning on a light). While a few implementations have been used, the UPnP functionality is primarily achieved through the use of a Java based library called Youpi that was developed as part of the Atraco project . This library has been released as open-source and is used in both living-labs (such as the LIMSI iRoom) and commercial products (such as those offered by inAccess networks).

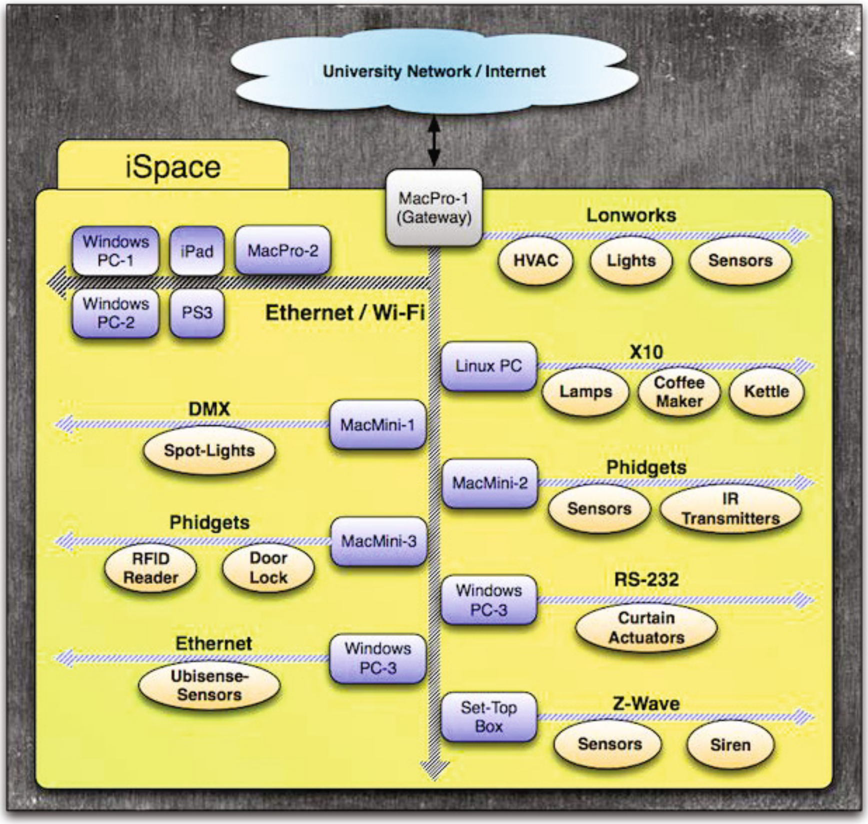


Fig. 7 Architecture of technology deployment within the iSpace

Although each gateway device in the iSpace is unique in its configuration, there are two main types. The first configuration uses a Java based OSGi framework that provides component management. Bundles of functionality (including the middleware) are deployed in this framework and can be done so dynamically during runtime. In this configuration a single runtime exists on the device and capabilities are added by installing bundles of functionality. This is easy to manage and efficient in operation due to the fact that only one instance of the middleware is running per device (and so only one middleware runtime needs to communicate over the network). In the second configuration, each component of functionality is wrapped in its own application. This requires more effort to manage and is less efficient from the perspective of the middleware, but is necessary where multiple components need to be deployed to a device and each component has been developed using different languages / tools. For example, our Windows PC-3 has two applications deployed on it (as shown in Fig. 7):

1. A Java application that advertises functionality to control Curtains and also provides a management GUI. This application operates its own instance of the middleware, has the control logic in-built and communicates via a RS-232 serial connection to the curtain actuators.
2. A C++ application that advertises functionality for the Ubisense real-time location tracking system (RTLS). This application operates its own instance of the middleware that wraps the installed software system (a Windows based application) through a C++ API.

While some of the bundles / applications wrap functionality by communicating directly with microcontrollers (as is the case for the curtains, lights, Phidget sensors / actuators and X10 devices) others exploit programming APIs of other software packages (such as the Ubisense RTLS and the Lonworks sensors / actuators), or have the exposed functionality in-built (such as is the case for the HTML5 based user interfaces and media repositories). The technical details regarding each implementation is beyond the scope of this chapter and could be realistically achieved using several approaches - what should be noted is the functionality they provide and the way in which they can be utilised. For example it is important to understand that a light can be advertised, described, manipulated and inspected through its software representation that is made available to agents / applications that are distributed across the network. In this specific example, three separate lighting technologies are used in parallel throughout the iSpace - but to a software consumer, a light of each type is indistinguishable from a light of any other type as the interfaces they implement are the same. The heterogeneity of the numerous components in the iSpace is homogenised through the middleware, allowing a single consistent model to be used.

The iSpace is an excellent experimental facility for multi-disciplinary research, this is especially useful across the spectrum of UC investigations where there is a symbiotic relationship between computer-science and social-science; two of the demonstrations from its portfolio are described below to holistically illustrate the concepts introduced through this chapter: *"FollowMe"* and *"HotSpot"* . Both of these demonstrators make use of RFID technology to recognise user-initiated events - a form of HCI that permits explicit user control. The experimental setup is the same for both the demonstrators and is shown in Fig. 8 below.

In particular, a multi-agent approach is adopted and the deployment consists of the following entities that are used to achieve desired operation:

1. **Real Entities**

   a. **Spot-Lights:** Eight dimmable spot lights are embedded in the ceiling of the living / kitchen area. Each is individually represented by a single UPnP device and can have its state (on / off) and intensity (0-100) controlled.

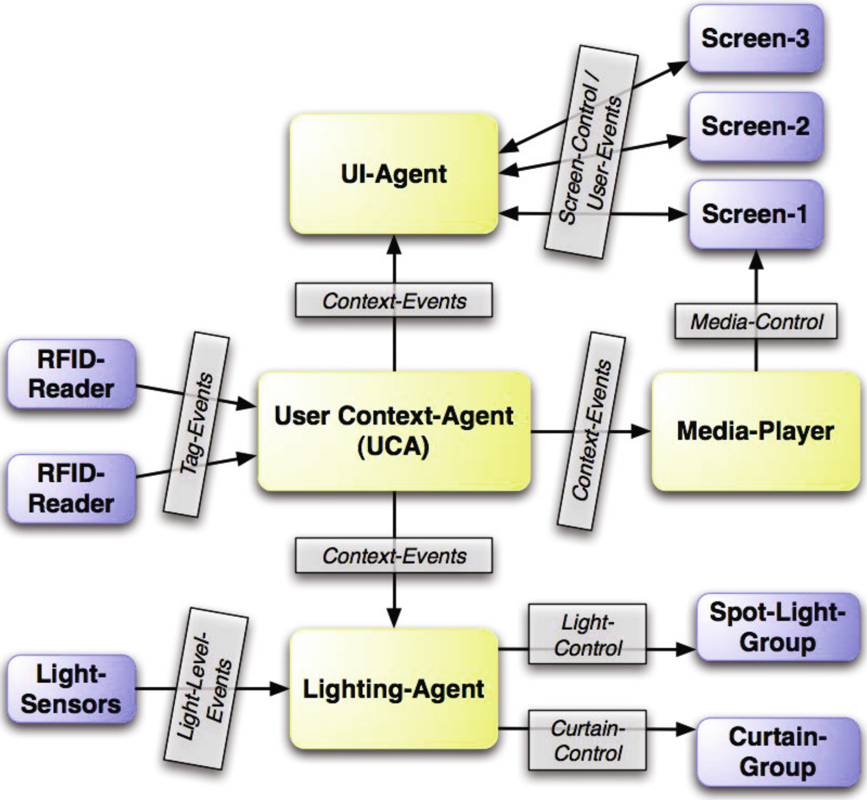


Fig. 8 Experimental setup of entities within the iSpace living-area for both FollowMe and HotSpot

b. **Light-Level Sensors:** Light sensors embedded in the walls and ceiling of the iSpace provide localised measurements for light-levels and can be used together to build a picture of the overall lighting conditions in the space. Values from each sensor can be retrieved through action invocation or by subscribing to the UPnP device for asynchronous event notifications.
c. **Curtains:** Two windows are equipped with motorised curtain controls. Each window is represented as a single UPnP device and can be in one of two states: OPEN or CLOSED.
d. **RFID-Readers:** Two RFID readers are deployed - embedded into the furniture of two contextual zones (markers indicate where a user must *“tag-in”*).
e. **Screens:** Of the six screens available, three are used within these demonstrators. Each shows a full-screen HTML5 web-browser that is connected to the UI agent (each screen provides a unique ID to the UI agent through the URL that it GETs). The HTML5 web-sockets feature is used to maintain a

bi-directional link with the UI agent - this allows events to flow asynchronously in both directions.

   i. **Screen-1:** A 40” LCD TV with a touch-sensitive overlay. This screen is used as a user interface.
   ii. **Screen-2:** A 40” wall-mounted Plasma screen used as an ambient media display.
   iii. **Screen-3:** A table-top projection (top-down LCD projector onto kitchen table) that provides user interface through a wireless trackpad.

2. **Virtual Entities**

   a. **Light-Group:** The eight *“real”* spot-lights can be addressed / controlled through a single UPnP device that represents them as a virtual group. Virtual light groups also exist for each of the contextual zones, but are not utilised in FollowMe / HotSpot.
   b. **Curtain-Group:** In a similar way to the Light-Group; the Curtain-Group provides a single and convenient representation for the two curtain devices to be treated as one.

3. **Logical Entities**

   a. **User Context-Agent (UCA):** This software agent has knowledge of fixed RFID reader locations *a priori*. It also has a database of known-users, each with an associated identity, RFID-tag and profile. A subscription to the two UPnP RFID readers allows RFID events to be monitored (the location of the event can be inferred by using the ID of the source RFID-Reader). A UPnP interface allows other agents to access user profile information and to subscribe for user context changes that are initiated when a user *“tags-in”* to a zone. Non-user RFID tags can be registered within a user-profile to generate specific events to subscribers (this is discussed further in the FollowMe and Hotspot sub-sections).
   b. **Lighting-Agent:** This Fuzzy-Task Agent (FTA) [55] controls the spot-lights to achieve lighting adaptation in response to context events from the UCA. A subscription to light-level sensors provides feedback from the environment and a fuzzy membership function is used in conjunction with the learned user preferences for light-levels.
   c. **UI-Agent:** This software agent has knowledge of fixed screen locations *a priori* and provides a HTML5 web-server for each screen to connect to (each screen provides a unique screen ID when it connects). This agent subscribes to the UCA and will modify the content of each individual screen when a user context change event is received.
   d. **Media-Player:** A UPnP media player is used to render audio and video on demand. When active it occupies full-screen on Screen-1 (replacing other active screen content such as the environment UI). For simplicity, the audio is simply output through the TV speakers, but could be direted to some other UPnP audio renderer if desired.

Why are there virtual groupings for lights / curtains? There are two reasons for this: *firstly*, it is more convenient and robust to develop software that deals with one remote resource than many. *Secondly*, Action invocation over a network using middleware has an inherent problem - it incurs a time overhead. More specifically, the current open-source *"Youpi"* implementation used requires ∼100ms to invoke an action on a UPnP device. And so, when an agent / application wants to achieve something like *turn on a bunch of lights*, there is a perceivable delay between the first and last light illuminating. Using a singly addressable group removes this problem as only one action invocation is required and so the individual lights respond together.

5.1 FollowMe

In this demonstrator, the user-interface for a specific occupant will migrate from screen-to-screen as he / she roams through the iSpace - thus it follows the user [46]. This relies on knowledge of screen locations (which are fixed) and user location (which is dynamic).

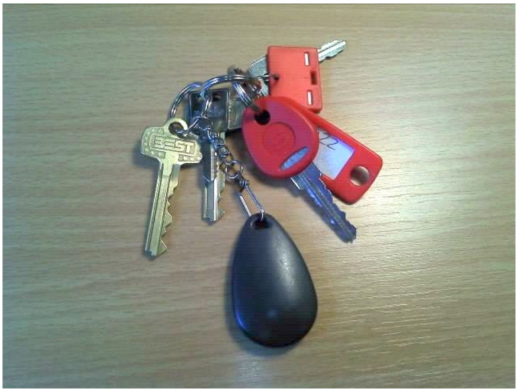


Fig. 9 RFID-tag attached to the keys of a user

When a user touches an RFID-tag (such as the one shown in Fig. 9) onto a reader, the UCA attempts to match the identity to one in its database. If the tag matches one registered as a user-context-tag, then a *context-change* event is distributed to all subscribed listeners. This event consists of the user-ID, a timestamp and a location-ID (inferred from the RFID-reader-ID). If the user is not already *logged-in* to the space, the listeners will utilise the user profile (available from the UCA) to configure certain aspects of the environment. The Lighting-Agent sets the appropriate light level and curtain state, the media player stops any currently playing media (and may start some background music if the user-profile specifies this preference), the UI-Agent transfers the UI to the screen closest to the user location (Screen-1 or Screen-3) and then sets the artwork on Screen-2 to the user preference.

A textual message is also popped-up on Screen-2 so that the user is informed of what just happened.

As the user roams through the iSpace, they can touch-in to other locations - this prompts the UCA to generate a new *context-change* event to all subscribed listeners. When the UI-Agent receives this event it will migrate the UI from whichever screen it is currently on and transfers it to the screen at the new user location.

5.2 HotSpot

In this demonstrator, a user can explicitly express some wish to the iSpace by placing an RFID-tagged object onto a reader. Fig. 10 shows three kinds of tagged objects - a document, some DVDs and a toy duck. The effect of each object on the space is specified in the user profile - it should be noted that a single object can therefore have a different effect depending on which user is currently logged-in. For the purpose of this discussion, we will present one of the authors profiles (physical document maps to digital document, DVDs map to movies and toy duck maps to music).

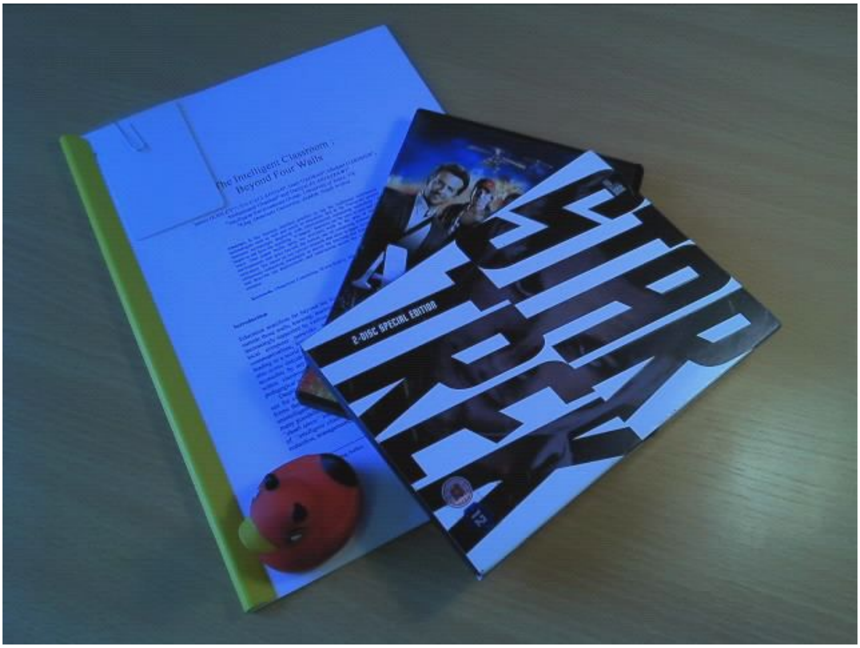


Fig. 10 RFID-tagged objects: a) a document, b) two DVDs, c) a toy duck

When a tagged object is placed on the reader, the UCA attempts to match the identity to one in the current user profile. Described below are the effects of three kinds of entity that are linked:

1. **Document:** A *context-change* event is generated by the UCA to indicate the user-activity is *"reading"*. The Lighting-Agent adapts the lights and curtains to the user preference for this task (for example all lights to full brightness and curtains open). The UI-Agent displays a digital form of the document (PDF) on Screen-2 and the Media-Player stops any current video / music.
2. **Movie:** A *context-change* event is generated by the UCA to indicate the user-activity is *"watching-movie"*.The Lighting-Agent adapts the lights and curtains to the user preference for this task (for example all lights to low brightness and curtains closed). The UI-Agent displays coverart on Screen-2 while the Media-Player goes full-screen on Screen-1 and then begins playback of the movie (streamed from a URL source over the network).
3. **Music:** Here the Genre of the linked music playlist is examined by the UCA in order to further specify the generation of a *context-change* event. As a result, the UCA notifies its listeners to indicate a *"relaxing-with-music"* activity. The Lighting-Agent adapts the lights and curtains to the user preference for this task (for example all lights to low brightness and curtains closed). The UI-Agent displays coverart on Screen-2 while the Media-Player goes full-screen on Screen-1 and then begins music playback (streamed from a URL source over the network).

6 Conclusions and Future Challenges

In this chapter we have discussed the convergence of IoT and IE approaches into a workable model that caters to both active and passive objects. A case study of the University of Essex iSpace is provided in which two demonstrators are described that illustrate the discussed convergence.

The abstraction of all things as entities allow software populations to make use of common functionality in order to reason with and manipulate a vast array of *"things"*. And so, the entities that form the fabric of an IE can be tailored to the preferences of a user. This is facilitated by middleware that renders a homogeneous distributed system from heterogeneous *"things"*. To ease the need on humans for direction and orchestration, intelligent agents collectively form an AmI that interacts with the real world through the use of sensors and actuators - providing a quality of intelligence that achieves autonomy.

The grand vision we have is for a world of Spaces, where each space constitutes an IE. People will be able to roam from Space-to-Space and enjoy a continuity of experience. There is still a lot of work to be done to achieve this. In particular, the inter-Space and intra-Space relationships need to be integrated and aligned to allow universal and global interoperability. The work towards realising AmI must also make breakthroughs, particularly in the support of multiple-users from its current proof-of-concept state in which single user scenarios are the norm. Security and privacy must also see a vast improvement before widespread adoption is made; perhaps the grandest challenge of all however, is to address the legal and societal boundaries to acceptance. The world is certainly becoming more accepting of technology in society - mobile phones, set-top boxes, tablet computers, etc. are already pervasive and have broken down the *"digital divide"* that once excluded certain groups of people from adoption. However, the more exciting current trend is the emergence of a global community that includes hobbyists and professionals alike that are in particular taking advantage of cheap and freely-available electronics such as [56] [57] [58] to realise new and novel creations that contribute to the IoT - an exemplar of technology in society.

What next could we envision once we have this world of Spaces ? perhaps Spaces that are structurally reconfigurable - a challenge more for architects and engineers within this, a multi-disciplinary field.

Acknowledgements. We gratefully acknowledge the support of HEFCE, the EPSRC, EU Framework Programmes, King Abdulaziz University, and the numerous other sources of funding that the Intelligent Environments Group has attracted over the past 10 years.

References

1. Weiser, M.: The computer for the twenty-first century, pp. 94–104. Scientific American (September 1991)
2. Brock, D.: The networked physical world, proposals for engineering the next generation of computing, commerce and automatic-identification. Auto-ID Centre Whitepaper (January 2001)
3. Magrassi, P.: A world of smart objects: The role of auto identification technologies. Strategic Analysis Report, Gartner (2001)
4. Brock, D.: The physical markup language. Auto-ID Centre Whitepaper (February 2001)
5. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., Spasojevic, M.: People, places, things: Web presence for the real world. In: 2000 Third IEEE Workshop on Mobile Computing Systems and Applications, pp. 19–28 (2000)
6. Barton, J., Kindberg, T.: The challenges and opportunities of integrating the physical world and networked systems. Technical report, HPL Technical report HPL-2001-18 (2001)
7. Want, R., Russell, D.M.: Ubiquitous electronic tagging. IEEE Distributed Systems Online 1(2) (February 2000)
8. Quack, T., Bay, H., Van Gool, L.: Object Recognition for the Internet of Things. In: Floerkemeier, C., Langheinrich, M., Fleisch, E., Mattern, F., Sarma, S.E. (eds.) IOT 2008. LNCS, vol. 4952, pp. 230–246. Springer, Heidelberg (2008)
9. Rekimoto, J., Ayatsuka, Y.: Cybercode: Designing augmented reality environments with visual tags. In: Proceedings of the ACM Designing Augmented Reality Environments (2000)
10. Adelmann, R., Langheinrich, M., Floerkemeier, C.: Toolkit for bar code recognition and resolving on camera phones - jump starting the internet of things. In: Proceedings of Workshop Mobile and Embedded Interactive Systems (MEIS 2006) at Informatik 2006, pp. 366–373 (2006)
11. Collins, D.J., Whipple, N.N.: Using Bar Codes – why it's taking over. Data Capture Institute (1990)
12. Falas, T., Kashani, H.: Two-dimensional bar-code decoding with camera-equipped mobile phones. In: Fifth Annual IEEE International Conference Pervasive Computing and Communications Workshops (PerCom), pp. 597–600 (2007)

13. Kindberg, T., Barton, J.: A web-based nomadic computing system. Computer Networks, Special Edition on Pervasive Computing 35(4), 443–456 (2001)
14. Falcao, V., Gibbons, J., Want, R., Hopper, A.: The active badge location system. ACM Transaction on Information Systems 10(1), 91–102 (1992)
15. Ward, A., Jones, A., Hopper, A.: A new location technique for the active office. IEEE Personal Communications 4(5), 42–47 (1997)
16. Balakrishnan, H., Priyantha, N.B.: The cricket indoor location system: Experience and status. In: Proceedings of the 2003 Workshop on Location-Aware Computing, Held in conjunction with UbiComp 2003, pp. 7–9 (2003)
17. Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., Balazinska, M., Borriello, G.: Building the internet of things using rfid: The rfid ecosystem experience. IEEE Internet Computing 13(3), 48–55 (2009)
18. Cadman, J.: Deploying commercial location-aware systems. In: Proceedings of the 2003 Workshop on Location-Aware Computing, Held in conjunction with UbiComp 2003, pp. 4–6 (2003)
19. Garcia-Valverde, T., Garcia-Sola, A., Hagras, H., Dooley, J., Callaghan, V., Botia, J.A.: A fuzzy logic based system for indoor localisation using wifi in ambient intelligent environments. To appear in IEEE Transactions on Fuzzy Systems (2012)
20. Rellermeyer, J.S., Duller, M., Gilmer, K., Maragkos, D., Papageorgiou, D., Alonso, G.: The Software Fabric for the Internet of Things. In: Floerkemeier, C., Langheinrich, M., Fleisch, E., Mattern, F., Sarma, S.E. (eds.) IOT 2008. LNCS, vol. 4952, pp. 87–104. Springer, Heidelberg (2008)
21. Haller, S.: The things in the internet of things. In: Procceedings of the Internet of Things Conference 2010, Tokyo, Japan (2010)
22. de Ruyter, B., Aarts, E.: Ambient intelligence: visualizing the future. In: Proceedings of the Working Conference on Advanced Visual interfaces (AVI 2004), pp. 203–208 (2004)
23. Johanson, B., Fox, O., Winograd, T.: The interactive workspaces project: Experiences with ubiquitous computing rooms. IEEE Pervasive Computing 1, 67–74 (2002)
24. Hamchi, I., Degré, S., Diduch, L., Rose, T., Fiscus, J., Fillinger, A., Stanford, V.: Middleware and metrology for the pervasive future. IEEE Pervasive Computing Mobile 8and Ubiquitous Systems 8(3), 74–83 (2009)
25. Ressel, C., Ziegler, J., Naroska, E.: An approach towards personalized user interfaces for ambient intelligent home environments. In: 2nd IET International Conference on Intelligent Environments, IE 2006, vol. 1, pp. 247–255 (2006)
26. Bellik, Y., Jacquet, C.: From the intelligent room to ambient intelligence. In: 1st Digiteo Annual Forum, Poster no. 2008-01 (2008)
27. Mozer, M.C.: The neural network house: An environment that adapts to its inhabitants. In: Coen, M. (ed.) Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, pp. 110–114 (1998)
28. Kidd, C.D., Orr, R., Abowd, G.D., Atkeson, C.G., Essa, I.A., MacIntyre, B., Mynatt, E.D., Starner, T., Newstetter, W.: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In: Yuan, F., Hartkopf, V. (eds.) CoBuild 1999. LNCS, vol. 1670, pp. 191–198. Springer, Heidelberg (1999)
29. Broxvall, M., Seo, B.S., Kwon, W.Y.: The peis kernel: A middleware for ubiquitous robotics. In: Proceedings of the IROS 2007 Workshop on Ubiquitous Robotic Space Design and Applications (2007)
30. El-Desouky, B., Hagras, H.: An adaptive type-2 fuzzy logic based agent for multi-occupant ambient intelligent environments. In: Proceedings of the 5th International Conference on Intelligent Environments, IE 2009 (2009)

31. Cook, D.J., Youngblood, M., Heierman, E., Gopalratnam, K., Rao, S., Litvin, A., Khawaja, F.: Mavhome: An agent-based smart home. In: Proceedings of the IEEE International Conference on Pervasive Computing and Communications, pp. 521–524 (2003)
32. Mattern, F., Cantero, M.O., Vidal, J.L.: Ubiquitous computing – the trend towards the computerization and networking of all things. Upgrade 2(5) (October 2001)
33. Streitz, N., Nixon, P.: The disappearing computer. Communications of the ACM 48(3), 32–35 (2005)
34. Kameas, A., Mavrommati, I., Markopoulos, P.: Computing in tangible: using artifacts as components of ambient intelligence environments. In: Ambient Intelligence: The evolution of Technology, Communication and Cognition, pp. 121–142 (2004)
35. Pounds-Cornish, A., Holmes, A.: The idorm - a practical deployment of grid technology. In: Second IEEE International Symposium on Cluster Computing and the Grid, CCGRID 2002 (2002)
36. Limb, R., Armitage, S., Chin, J., Kalawsky, R., Callaghan, V., Bull, P., Colley, M., Hagras, H.: User interaction in a shared information space - a pervasive environment for the home. In: Proceedings of the IEE Workshop on Perspectives in Pervasive Computing (October 2005)
37. Hagras, H., Callaghan, V., Colley, M., Clarke, G., Pounds-Cornish, A., Duman, H.: Creating an ambient-intelligence environment using embedded agents. IEEE Intelligent Systems 19(6), 12–20 (2004)
38. Rivera-Illingworth, F., Callaghan, V., Hagras, H.: A neural network agent based approach to activity detection in ami environments. In: Proceedings of the First IEE International Workshop on Intelligent Environments, IE 2005 (2005)
39. Dooley, J., Callaghan, H.H.V., Gardner, M., Ghanbaria, M., AlGhazzawi, D.: The intelligent classroom: Beyond four walls. In: Intelligent Campus Workshop (IC 2011) held at the 7th IEEE Intelligent Environments Conference, IE 2011 (2011)
40. Dooley, J., Ball, M., Al-Mulla, M.: Beyond four walls: Towards large-scale intelligent environments. In: Proceedings of the First Workshop on Large Scale Intelligent Environments (WOLSIE), col-located with the 8th International Conference on Intelligent Envrironments, IE 2012 (2012)
41. Cook, D.J., Das, S.: Pervasive computing at scale: Transforming the state of the art. Journal of Pervasive and Mobile Computing 8(1), 22–35 (2012)
42. Beslay, L., Hakala, H.: Digital territory: Bubbles. In: European Visions for the Knowledge Age: A Quest for New Horizons in the Information Society (2007)
43. Whittington, L., Dooley, J., Henson, M., AL-Ghamdi, A.A.-M.: Towards followme user profiles for macro intelligent environments. In: Proceedings of the First Workshop on Large Scale Intelligent Environments (WOLSIE), col-located with the 8th International Conference on Intelligent Envrironments, IE 2012 (2012)
44. Dooley, J.: An Information Centric Architecture for Large Scale Description and Discovery of Ubiquitous Computing Objects. PhD thesis, University of Essex (2011)
45. Dooley, J., Henson, M., Callaghan, V., Hagras, H., AlGhazzawi, D., Malibari, A., AlHaddad, M., AL-Ghamdi, A.A.-M.: A formal model for space based ubiquitous computing. In: Intelligent Environments Conference, IE 2011 (2011)
46. Dooley, J., Wagner, C., Hagras, H., Pruvost, G.: Followme: The persistent gui. In: 1st International Workshop on Situated Computing for Pervasive Environments (SCOPE 2011) at the 6th International Symposium on Parallel Computing in Electrical Engineering, PARELEC 2011 (2011)
47. Jin, Y., Wang, R., Huang, H., Sun, L.: Agent-oriented architecture for ubiquitous computing in smart hyperspace. Wireless Sensor Network 2, 74–84 (2010)

48. Jammes, F., Mensch, A., Smit, H.: 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, MPAC (2005)
49. Zeeb, E., Bobek, A., Bohn, H., Golatowski, F.: Service-oriented architectures for embedded systems using devices profile for web services. In: 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW), vol. 1, pp. 956–963 (May 2007)
50. Emmerich, W.: OMG/CORBA: An Object-Oriented Middleware. In: Marciniak, J.J. (ed.) Encyclopedia of Software Engineering, pp. 902–907. John Wiley and Sons (2002)
51. Henning, M.: A new approach to object-oriented middleware. IEEE Internet Computing 8(1), 66–75 (2004)
52. Kon, F., Costa, F., Blair, G., Campbell, R.H.: The case for reflective middleware. Communications of the ACM 45(6), 33–38 (2002)
53. Chin, J., Callaghan, V., Clarke, G.: Soft-appliances: A vision for user created networked appliances in digital homes. Journal of Ambient Intelligence and Smart Environments, JAISE (2009)
54. Keith Edwards, W., Newman, M.W., Sedivy, J.Z.: The case for recombinant computing (technical report csl-01-1). Technical report, Xerox Palo Alto Research Center (2001)
55. Bilgin, A., Dooley, J., Whittington, L., Hagras, H., Henson, M., Wagner, C., Malibari, A., AlGhamdi, A., AlHaddad, M., AlGhazzawi, D.: Dynamic profile-selection for zslices based type-2 fuzzy agents controlling multi-user ambient intelligent environments. In: Proceedings of the 2012 FUZZ-IEEE Conference (2012)
56. Callaghan, V.: Buzz-boarding; practical support for teaching computing, based on the internet-of-things. In: 1st Annual Conference on the Aiming for Excellence in STEM Learning and Teaching (April 2012)
57. Upton, E., Halfacree, G.: Raspberry Pi User Guide. John Wiley and Sons (2012)
58. Buechley, L., Eisenberg, M., Catchen, J., Crockett, A.: The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In: Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, CHI 2008, pp. 423–432. ACM (2008)

Generalized World Entities as an Unifying IoT Framework: A Case for the GENIUS Project

Gian Piero Zarri

Abstract. After having briefly discussed some possible interpretations of the (still at least partially ambiguous ambiguous) "IoT" term, this Chapter sums up the aims and the main characteristics of an on-going IoT-inspired project, GENIUS. GENIUS concerns the creation of a flexible, internet-based, IoT cognitive architecture, able to support a wide range of 'intelligent' applications focused on the recognition and interaction with the so-called Generalized World Entities (GWEs). The GWE paradigm intends to fill up the present fracture between the detection of entities at the sensor/physical level and their representation/management at the conceptual level. It deals in a unified way with physical objects, humans, robots, media objects and low-level events generated by sensors and with GWEs at higher level of abstraction corresponding to complex, structured events/situations/behaviours implying mutual relationships among GWEs captured at lower conceptual level. GWEs of both classes will be recognised and categorised by using, mainly, a conceptual "representation of the world", ontology-based, auto-evolving and general enough to take into account both the "static" and "dynamic" characteristics of the GWEs. When all the GWEs (objects, agents, events, complex events, situations, circumstances, behaviours etc.) involved in a given application scenario have been recognised, human-like reasoning procedures in the form of "set of services", general enough to be used in a vast range of GWE-based applications, can be used to solve real-life problems. Details about the use of the GWE paradigm to set up an "Ambient Assisted Living (AAL)" application for dealing with the "elderly at home problem" are provided in the Chapter.

Keywords: Generalized World Entities, IoT, Ontologies, Sensor Level, Inferences.

1 Introduction

This Chapter describes the general aims of the GENIUS (**GEN**eral**I**zed world entities, a **U**nifying iot **S**upport) project and outlines the technical/scientific procedures already specified (and partially implemented) in the framework of this project. Based on previous (European) initiatives and experiments, GENIUS is

Gian Piero Zarri
Sorbonne University, LaLIC/STIH Laboratory,
Maison de la Recherche, 28 rue Serpente, 75006 Paris, France
e-mail: zarri@noos.fr, gian_piero.zarri@paris-sorbonne.fr

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 345–367.
DOI: 10.1007/978-3-642-34952-2_14

presently led and carried on and by staff associated with the LaLIC/STIH Laboratory of the Sorbonne University in Paris, France.

Besides its (actual and future) *concrete* achievements, see next Sections, GENIUS would also like to contribute to the *theoretical* developments of the IoT domain by introducing some clarifications about the object of study of this discipline.

As well-known, in fact – and in spite of the early inclusion of IoT, by the US National Intelligence Council (NIC) [1], among the six “disruptive civil technologies” with potential impacts on US Interests out to 2025 – a stable and universally accepted definition of the domain of interest covered by the “IoT” term is still lacking. For example, as a heritage of the first years of existence of this discipline (strongly influenced by the RFID technology [2]) and according to a quite reductive physicalist approach, the “T” of “IoT” is systematically assumed to mean merely “*physical* Things”. Let us look, e.g., to this passage in the introductory Chapter of a recent book on the architecture of IoT – in all the subsequent citations of this Section, “italics” means “emphasis added”: “A minimalist approach towards a definition may include nothing more than things, the Internet and a connection in between. Things are any *identifiable physical object* independent of the technology that is used for identification or providing status information of the objects and its surroundings…” [3: 8]. Not too different is the position expressed in the beginning of Atzori and colleagues’ survey paper: “The basic idea of this concept (IoT) is the pervasive presence around us of a variety of things or objects – *such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc.* – which, through unique addressing schemes, are able to interact with each other …” [4: 2787]. However, other definitions evoke also the existence, in a IoT context, of “*virtual things*” along with the previous “*physical objects*”, see Santucci, “…It (IoT) refers to a world where physical objects and beings, *as well as virtual data and environments*, all interact with each other in the same space and time” [2: 4] or the CERP-IoT (CERP = Cluster of European Research Projects) definition: “…Internet of Things … is defined as a dynamic global network infrastructure with self-configuring capabilities … *where physical and virtual ‘things’ having identities, physical attributes, virtual personalities and using intelligent interfaces are seamlessly integrated into the information network*” [5: 6]. Unfortunately, some ambiguities subsist when trying to determine which sort of entities correspond exactly to the “virtual things”. In the majority of cases, these virtual entities seem simply to denote a sort of *Internet image* of the ordinary physical things bringing in some additional information see, in this context, this statement that appears frequently within the Chapters of the book on the architecture of IoT mentioned above: “The Internet of Things *links uniquely identifiable things to their virtual representations in the Internet*” [6: 253]. According to a possible “ontological” interpretation, these virtual things could then correspond to those “*concepts*” that supply a generalized ‘abstract’ view of all the possible low-level ‘identifiable’ entities, see also Section 3.2 below. But we can also find traces in the literature of “virtual things” endowed with proper and independent characteristics and corresponding to events, services, actions or, as in [7: 145], to “*immaterial logistics objects*”:

"… immaterial logistics objects can be considered to be Intelligent Products with intelligence located in the network, *but without a physical manifestation* … For instance, orders, invoices …". In these last cases it is unclear, however, why an order should be "immaterial" and why the event concerning issuing or satisfying this order should be "virtual".

The GENIUS project capitalizes on the contrary on the so-called *Generalized World Entities (GWEs) paradigm.* This last makes use of a *unique conceptual formalism* to, on the one hand, overcoming the persistent "physicalist" tendencies still affecting the IoT domain and, on the other, compensating for the present, limited abilities of the IoT-like techniques to generalize from observed situations, to adapt to new, dynamic contexts, or to take care of human intentions/behaviours. The formalism should then be able to deal in a *standardised and uniform way* not only with (known and unknown) *physical objects*, but also with (human and mechanical) agents, events, situations, circumstances, behaviours etc. and their evolution in time, as well as with the relationships between all these *generalized world entities.*

In the following, Section 2, we will first explain in some depth the GWE concept by using a detailed example. Section 3 will describe the operational procedures (identification and low-level description of the GWEs, categorization, reasoning based on their full recognition, "model of the world" etc.) allowing us to make concretely use of the GWE paradigm. Section 4 will consist in the description of the use of GWEs in the context of an "Ambient Assisted Living (AAL)" real-life scenario. Section 5 is a short "Conclusion".

2 The GWE Paradigm

2.1 General Context

Independently from the ambiguities about the exact limits of the IoT domain discussed above, there is a growing consensus about the need of introducing some "deep reasoning" capabilities into the artefacts/procedures actually used in this domain. This will allow them to go beyond their (relatively limited) present possibilities of taking given situations into account, of smoothly adapting to their environment and of operating in unforeseen circumstances. Current solutions must often be tailored to specific tasks and environments and still require lengthy human supervision. We would like, on the contrary, to make use of artefacts/procedures able to deal autonomously and in a somewhat 'intelligent' way with complex activities like planning, monitoring, recognition of complex environments, reasoning, detection of intentions/behaviours, etc. – all tasks that, in one way or another, pertain surely to the IoT domain.

At the same time, a whole panoply of tools like advanced knowledge representation systems, powerful inference techniques, semantic-based advanced services etc. – that could represent, at least in principle, a suitable solution for adding more 'intelligence' to the IoT-like techniques – has been developed in a cognitive science community context. *The set up of 'bridges' between the two environments would then appear both a logical and useful step.*

The implementation of this merged approach implies, however, solving some difficult problems, technical problems first. To give only an example, the adoption of this 'merge' implies, among other things, the capacity of 'translating' into a conceptual model, making use of ontologies and rules, the outputs of those sensor's systems that anyway represent, in the IoT domain, the privileged way of acquiring data from the environment. Given the theoretical and practical divergence between these "signals" – often in numerical form – and the symbolic "concepts" of the cognitive approach, this passage represents an obstacle particularly uneasy to surmount. Moreover, there is also a sort of 'working environment gap' between the two communities, that of the "IoT scholars" and that of the "cognitive scholars": the tools they utilize are quite different (algorithmic and probabilistic techniques on one side, ontological engineering on the other) and used normally for quite different purposes.

Several teams all around the world are already working in this 'bringing together' framework, with some interesting results obtained, for example, in the IoT-related domain of "cognitive robotics" [8, 9, 10]. To enhance intelligence of robots in complex real-world scenarios, recent research trends aim, in fact, to introduce high-level semantic knowledge in different functions of robots such as: mapping and localization, scene understanding, smart and seamless interactions with human and environment for ambient assisted livings and ubiquitous spaces.

As already stated, the GENIUS project is built around the notion of "Generalized World Entities" (GWEs). From a concrete point of view, GWEs can consist of physical objects, humans, robots, of media objects like text, images, video and speech, and of low-level 'events' where information is generated by sensors. In addition, GWEs can also represent *entities at a higher level of abstraction* – generalisations, in a sense, of the "virtual things" discussed in Section 1 – such as complex, structured events/situations/behaviours/circumstances originated from the mutual interactions and relationships among GWEs captured at a lower conceptual level see, e.g., the examples of complex GWEs originated in an AAL context reproduced in Table 1 below, Section 4. GWEs are then characterized by the presence of extensive spatio-temporal information. The GENIUS project involves then the main following aspects:

- The use of advanced methods and formalisms to recognize, describe and reason about the occurrences of "Generalized World Entities" (GWEs) that, as already discussed, are not limited to 'physical things' but provide a uniform framework to represent objects, human or mechanical agents, events, situations, behaviours and their evolution in time, as well as their relationships.
- The conception and implementation of tools for creating and maintaining the links between the descriptions of the GWEs at conceptual (ontological) level and their concrete instances, as these last are perceived from the real world through the use of any possible sort of video, audio, RFID etc. sensors. Note that the possibility of realising an easy, uniform passage from the "sensor level" to the corresponding" conceptual level" is crucial for filling the gap mentioned above, and it represents the main difficulty to be overcome in this context.

- The development of reasoning (inference) procedures under the form of a ‘set of services’ (for accident avoidance, monitoring, planning, expectation/ discovering of possible behaviours etc.), general enough to be used in a vast range of GWE-based applications and that do not need to refer to any previous, lower-level recognition effort.
- The validation of the resulting cognitive architecture through the implementation of real-world scenarios (proofs of concepts), able to demonstrate the versatility and general-purpose nature of the proposed approach.

To attain its objectives, the GWE paradigm makes use of several conceptual tools, relatively unusual in a ‘standard’ IoT environment; they are, mainly:

- A conceptual “*representation of the world*” – built up, at least partially, on the basis of already existing tools – that is ontology-based and auto-evolving (see further details in 3.2 below) and is general enough to take into account both the ‘static’ and ‘dynamic’ characteristics of the (known and unknown) GWEs. The representation must be able to deal *in a unified way* with objects, human beings, robots, events, situations, circumstances, behaviours etc. and their relationships: all these entities – the last in particular – are handled in fact as *first class citizens* in the framework of the ontological formalism.
- A set of *inference rules* – and the corresponding inference engines – that work at different levels of granularity and that are able to:
  - Solve knotty cases concerning the correct description of the features initially associated, at the ‘physical’ level, with the GWEs through the use of sets of sensors in particularly complex events/situations.
  - Find a correspondence between these low-level features and the high level conceptual descriptions included in the world representation by taking into account all the incoming GWEs through on-line updating.
  - Implement the reasoning techniques to be used to infer all the possible consequences/suggestions for actions that can be derived from having completely recognised/described a given (complex) event/situation.

2.2 A First Example

To explain simply what “GWE” means in practice and, at the same time, clarify the general philosophy of the GENIUS project, let us consider a partially blind senior (or a baby) who is moving in an environment where there is an object along their projected path. The object can be a dangerous obstacle, e.g., a table, or a relatively harmless one, e.g., a newspaper or a puddle.

To fully understand this (highly schematised) situation (see Fig. 1), *three independent GWEs must be recognized and dealt with*: the “*person*”, the “*object*” and a “*MOVE complex event*” where the GWE corresponding to the “person” fills the *subject/actor/agent “role”* and the GWE corresponding to the “object” (the obstacle) fills the *location/direction “role”*. From an ontological point of view, the first two GWEs correspond to instantiations of ‘standard’ concepts like human_being and physical_object, whilst the third GWE corresponds to the instantiation of a different (and more complex) type of conceptual entity that is used to represent the *(dynamic) events/situations/circumstances/behaviours arising from the interrelationships of the previous, lower level entities*. In this last case, conceptual notions like that of "*semantic predicate*" (e.g., MOVE) and "*functional role*" (e.g., SUBJECT) must be used. All along this Chapter, the terms set in Arial font will refer to the high-level conceptual representation language used in GENIUS, i.e., NKRL (Narrative Knowledge Representation Language), see [11, 12] and Section 3.2 below. More precisely, the Arial terms including "underscore" characters denote concepts, like "human_being" (lowercase letters), or instances (individuals), like "JOHN_" (uppercase letters). Arial terms without underscores denote semantic predicates ("MOVE") or functional roles ("SUBJECT") when set in uppercase letters, formal descriptions of general situations ("templates" in NKRL terms) like Move:AutonomousPersonDisplacement, see below, when set in lowercase letters.

The first move to implement in order to model the situation of Fig. 1 consists now in i) recognizing the presence of the three GWEs, ii) assigning automatically a provisional identifier (URI-like) to each of them and mainly iii) extracting a set of 'features' (in the most general meaning of this term) useful for their initial characterization. To do this, standard techniques can be used, typically coupling a network of (wired or wireless) sensors of different types for detecting motion and localization – 'invasive' like cameras or GPS or 'less invasive' like RFID, contact switches and pressure – with signal processing algorithms capable of recognizing, e.g., edges, corners, interest points, curvatures etc. on the basis of a search for discontinuities. Note that the set of features associated in this way to each GWE is not yet the proper 'recognition' of the corresponding entity but only a sort of 'low-level description' built up from information provided by many different sensors. With respect, e.g., to the GWE corresponding to the movement of the person towards the obstacle, this movement can correspond simply, at this 'feature' level, to identify the presence of the same entity in two different positions.

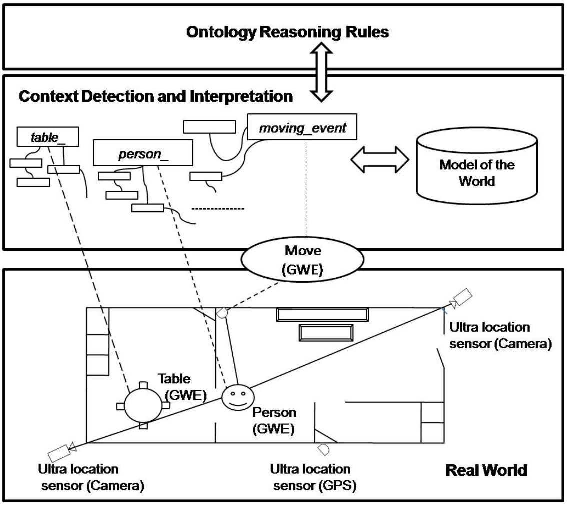

Ontology Reasoning Rules
Context Detection and Interpretation
table_
person_
moving_event
Model of the World
Move (GWE)
Ultra location sensor (Camera)
Table (GWE)
Person (GWE)
Ultra location sensor (Camera)
Ultra location sensor (GPS)
Real World

Fig. 1 A simple example illustrating GWEs' main features

The proper "recognition/categorization" of the three GWEs is carried out in a following phase, *by associating the three provisionally identified entities to the corresponding "conceptual descriptions" that are part of the auto-evolving general ontology of the GENIUS project*. This last provides the specific 'description of the world' that is appropriate for a particular running application. "Recognition" must then be understood here as "*conceptual recognition*", i.e., the fundamental phase that transforms the 'physical' description of the starting situation/event into a 'conceptual' (ontology-based) description of this situation/event/behaviour, authorising then the use of all the possible Artificial Intelligence, Cognitive Sciences, Semantic Web etc. querying/reasoning/inferencing tools.

The provisional identifiers attached to the three GWEs in the previous phase are then removed and changed into conceptual labels in the style of TABLE_1 (or PUDDLE_1, NEWSPAPER_1), AGEING_PERSON_1 (or BABY_1) and PERSON_DISPLACEMENT_1 to indicate the fact that they have now been categorized as instances of 'standard concepts' like table_, puddle_, ageing_person or baby_ and of 'dynamic situations/events/behaviours…' like Move:AutonomousPersonDisplacement. Basically, this recognition phase utilizes both i) standard machine learning techniques, syntactic or probabilistic (Bayesian) that make use of the collected features to identify a given entity, and ii) unification techniques for combining these features with those characterizing the general conceptual classes (table_, Move:AutonomousPersonDisplacement etc.) to which the GWEs will eventually be associated. Note that the representation of a conceptual entity (of an NKRL "template") like Move:AutonomousPersonDisplacement must be sufficiently complete and structured to be able to give rise to an instance – e.g., the GWE PERSON_DISPLACEMENT_1 – described as the MOVE of a "subject" as AGEING_PERSON_1/BABY_1 towards a "location" like TABLE_1/PUDDLE_1, and must then make use of advanced conceptual tools in the style of semantic predicates (MOVE), functional roles (SUJET), etc., see again 3.2 below.

Having recognized the unknown GWEs, the last step consists then in 'reasoning' about the global situation (by using an "extended production rules" approach, see 3.4) in order to implement the 'action plans' proper to the situation/event that has been recognized. In this simple example, we must then *infer* that, if the obstacle is a puddle or a newspaper, an action to stop the movement of the elderly person or the baby it is not necessary, but their movement must be certainly halted if the object is, for example, a table or a glass door.

The above scenario is very basic, and has been introduced only to show how a high-level GWE corresponding to a complex situation can be built up by aggregating GWEs of lower order in the framework of a coherent conceptual model. The scenario can, however, be easily generalized, e.g., by substituting a simple Pekee II Mobile robot – equipped with a Vision 3D+ system, a Pan-Tilt camera, an RFID Reader and an ambient sound sensor – to the elderly person/baby. The SUBJECT of the global MOVE will obviously change, as the ways of recognizing, at sensor level, the low-level GWEs, *but the general framework is still perfectly consistent.* Moreover, in a 'behavioural' perspective, a

GOAL GWE could be associated with the MOVE one in case it could be possible to discover that the movement is directed towards the kitchen or the toilet, etc.

3 The GENIUS' Procedures Implementing the GWE Paradigm

3.1 Identification and Low-Level Description of the GWEs

These procedures concern the detection and the accurate description/ characterization of all the possible Generalized World Entities (GWEs) coming from an external data stream (or from several streams) – static objects, persons, physical low-level events where information is generated by sensors, multimedia information (text, still images, video and video fragments, sounds etc.), but also spatio/temporally bounded events/situations/circumstances implying real time, mutual interactions among other GWEs. The original data stream(s) can be generated from a variety of different hardware-based sensors of different levels of complexity, including RFIDs, contact switches and pressure mats, cameras, LIDARs and radars, and Wireless Sensor Networks (WSNSs). Visual and infrared signals, and 3-D video tracking systems, can be used for detecting and tracking the presence, identity and activities of given entities; audio signals and speech recognition techniques can prove useful in order to classify the interactions between people, etc. This identification/description activity means that GWEs – in conformity with the usual IoT requirements see, e.g., [4] – must i) all – including those corresponding to complex situations/events – be characterized by a *(provisional) identifier* (URI-like), to be changed into a specific instance label in the following recognition/categorization phase; ii) be endowed with a *set of features/properties* to be calculated in real time; and iii) be supplied with an (at least elementary) *interface* allowing them to communicate/be integrated with other GWEs. With respect to this last point, for example, GWEs under the form of elementary events must be able to interact among them to give rise to complex events GWEs, to be recognised then as single units at the conceptual level.

Extracting the initial characteristic features (identifiable attributes/properties) of the GWEs by analysing the output of a network of sensors – for temperature, motion, localization (RFID, GPS…), weight etc. – is a complex activity that usually make use of several integrated techniques. For example, with respect to the 'pure physical' objects, all sorts of 'analytical' tools like first- and second-order derivative expressions, Haar transforms, auto-regressive models and Canny-Deriche etc. filters (for edge detection), local colour descriptors, global colour histograms, (syntactic) pattern recognition techniques and discriminant factor analysis (for identify movements) etc. can be used. "Crossed extraction" – i.e., using the outputs of other sensors to reinforce the analysis proper to a given sensor – can be employed to obtain more reliable results; use of COTS (Commercial Off-The-Shelf tools) can also, of course, be envisaged.

Detection and tracking of humans is of particularly high importance in a GENIUS framework. RGB-D sensors such as Kinect [13] and the model-based approaches for the estimation of 3D positions of body segments are considered to be currently the best solution for human body tracking. Nevertheless, clothing, poor lighting and other factors can cause the Kinect-based trackers to fail. In GENIUS, we are then actually investigating how to augment Kinect-based skeleton trackers with other types of trackers that might provide less complete information than full-fledged skeleton tracking but are less prone to failure in difficult situations. One possibility is to track each human body part separately, thus providing at least partial information about the position and configuration of human body parts: body part tracking can be based on parametric 3D models of body segments, e.g., based on different combinations of "*geons*" (simple 3-D geometric primitives for object representation) for the torso, the head, upper and lower arms, as well as upper and lower legs, which can be fitted into the candidate sets. The fitting has to deal with a large search space due to the necessary parameterisation of the geons to deal with the high variability of humans due to factors like age, gender, clothing styles, etc. Alternatively, statistical techniques like Viola-Jones approach [14] to face detection can be utilized to segment body parts. Furthermore, there can be an arbitrary number of humans in the scene; it is therefore necessary to arrange the fitting and classification in a hierarchical manner to allow for an efficient, real time capable detection and tracking of humans. The output of these simpler trackers will be integrated with the output of full-body skeleton trackers to ensure that an estimate of the human position will always be available, albeit with different granularity and degrees of certainty. Furthermore, the above detection and tracking based on RGB-D sensors must be complemented, in case of ambiguity, by qualitative tracking using simpler distributed sensors, including IR proximity sensors, pressure sensors, and door switches.

Note, to conclude, that the techniques mentioned in the previous paragraphs can be successfully used to identify 'single' GWEs and their characteristic features – e.g., by reconstructing the constitutive elements, edges, corners, interest points, curvatures etc. of a squared_object and signalling also the presence of an individual that is MOV(ing). Identifying, in contrast, all the characteristics of structured GWEs associated with complex events, situations and circumstances and their 'behavioural properties' – e.g., describing at the feature level the whole GWE representing an entity_ that is MOV(ing) towards a squared_object – requires, in the most complex cases, the execution of inference operations implying the use of the 'modelling of events and situations' component of the world model defined below.

3.2 Conceptual Representation of the World

According to the well-known Gruber's formulation [15], a consensus definition of "*ontology*" says that "Ontologies represent a formal and explicit specification of a shared conceptualization", where:

- *Conceptualization* refers to an *abstract model* of some phenomenon/situation in the world, where the model follows from the identification of the relevant "*concepts*" (i.e., the essential notions) that characterize this particular phenomenon/situation.

- *Explicit* means that the type of concepts used, and the constraints on their use, are *explicitly defined*.
- *Formal* refers to the fact that the ontology should be *computer-usable*.
- *Shared* reflects the notion that an ontology captures *consensual knowledge*, that is, this knowledge is not private to some individual, but must be accepted by a group.

An ontology represents, therefore, an *explicit, formal and consensual representation* (a "model of the world") of a given application domain. This model can be *shared* and, given its '*uniqueness*' properties, is also able to take the *interoperability problems* into account. In GENIUS, we make use of an ontological approach to provide a *unique conceptual representation* for all the possible GWEs proper to a given domain/application, independently from the fact that these GWEs correspond to physical objects, to virtual things, to simple or complex events, to structured behaviours/situations/circumstances etc. We can note here that an ontological approach to be used to describe the 'world' seems to be still quite unusual in an IoT context: see, however, [6, 7, 16, 17].

With respect now to the specific "*knowledge*" to be inserted into a GENIUS ontology it should appear clearly, from what stated in Section 1 and from the example of Fig. 1, that this knowledge must be classed into *three different but strictly related categories*:

- *"Plain/static" knowledge* corresponding to *stable, self-contained and basic notions* that can be considered, at least in the short term, as *'a-temporal' (static) and 'universal'*. This means, among other things, that their *formal definitions* within the ontology are not subject to change, at least within the framework of a given application, even if they can evolve in the long term as a consequence, e.g., of the progress of our knowledge. These static notions can be very general (corresponding then to surface lexical terms as "human being", "building" or "artefact") – and proper, then, to several application domains – or linked to specific application domains (like "home control system", "level of temperature", "valve", "emergency alarm" or even simple, low-level events like "button pushing"). We can note that these "static notions" are not, at least in principle, *necessarily restricted* to any limited "physical things" interpretation, see Section 1. fire_breathing_dragon – a specific term of dragon_ – is a standard concept whose definition is certainly *stable* in the context of any possible (fairy tales more than IoT) application and that can be legally 'materialized' into some instances (individuals) of the GLAURUNG_ type.
- *"Structured/dynamic" knowledge*, intrinsically *transient* and then *highly time-dependent*, which corresponds to the description of *dynamic events/ situations/circumstances* where the above "static" entities are involved – e.g., a robot or a blind person encountering a table during her/his/its movements, see Fig. 1 above, an impaired person triggering an emergency alarm or a surveillance system detecting an intrusion in a protected building (or my yesterday meeting with my dear friend Glaurung). The basic units of this second type of knowledge – called "*elementary events*" in knowledge

representation terms and corresponding, e.g., to "the robot moves towards the table" – are *highly structured pieces of information*. To be represented correctly they, and the corresponding GWEs, ask for at least the use of i) "*conceptual predicates*" in the style of, e.g., MOVE in the previous example for specifying the basic type of state, action, process etc. described in each elementary event, and ii) the notion of "*functional role*" [18] to denote the logical and semantic function of each of the "plain/static" entities (GWEs) involved in the event. For example, as already stated, a SUBJECT functional role must be used to specify the 'main' function of the GWE that is moving, and a DESTINATION role is needed for differentiating the function of the previous entity from that of its (known or unknown) 'target'. Moreover, an exhaustive representation of the *associated temporal phenomena* is required.

- A last sort of knowledge concerns the modelling of the so-called "*connectivity phenomena*", i.e., the coherence links that bring together into a unique *complex event* (a 'stream of events') its different, constitutive elementary event units. These links are normally expressed in natural language through syntactic constructions like causality, goal, indirect speech, co-ordination and subordination, etc., see the example: "The person/robot is moving *in order to* (GOAL) open the fridge or pass a door".

The formal structures needed for taking correctly into account these three different forms of knowledge are quite different. With respect to the plain/static knowledge, its *self-contained and stable character* – where the temporal phenomena can be ignored – justifies a conceptual representation/definition of the corresponding GWEs according to some *simple model like the traditional, "binary" one*. In this last approach, the "properties" or "attributes" that define a given "concept" are then expressed as *binary (i.e., linking only two arguments) relationships of the "property/value" type*, independently from the fact that these relationships are organised, e.g., into "frame" format as in the original Protégé model, see [19], or take the form of a set of "property" statements defining a "class" in a W3C (World Wide Web) language as OWL/OWL 2, see [20, 21].

While the standard (binary) ontologies and the W3C solutions (RDF/S, OWL, OWL 2 etc.) may be sufficient to represent correctly, under GWE form, the "plain/static" knowledge – note also the emergence of newer, up-to-date *binary proposals* in the "schema.org" style [22] – they are *conceptually inadequate* (or at least, *very inefficient from a practical point of view*) to represent the two residual kinds of knowledge (and the temporal information). For this type of information, in fact, the need for *structured, n-ary forms of knowledge representation* is now widely recognized see, among many others studies, [23, 24, 25]. In a GENIUS context, we deal then with the structured/complex/dynamic GWEs making use of tools based on the NKRL approach [11, 12], a language/environment characterised by the presence of *two different but strictly integrated ontologies*. The first, HClass (hierarchy of classes) allows for the representation of the "plain/static" knowledge according to the usual "binary" approach. The second, HTemp (hierarchy of templates) is used for the encoding of the "structured/dynamic" knowledge: the templates are n-*ary descriptions*, based on the notion of "semantic predicate" and "functional roles", of *general classes of elementary events* like "human beings/artefacts are moving forward", "forced displacement of a physical object", "production of a supporting service", "sending/receiving messages", "make a change of state happen", etc.

The GWEs corresponding to the concrete instantiation of templates, "(<R2D2_ is moving towards TABLE_27>)", can then be associated making use of *second order recursive labelled lists* to build up complex scenarios see, e.g., "(<R2D2_ is moving towards TABLE_27>) GOAL (<R2D2_ will pick up CUP_32>)" taking, therefore, the "*connectivity phenomena*" knowledge into account. More exactly, the second order lists used in this last context include *the symbolic labels of the original elementary events ("reification")*, making then (*indirect*) reference to the representations of these events; see also, in this context, [11: 91-98] and the analysis of the AAL scenario in Section 4 below.

A detailed description of the procedures used in NKRL to deal with temporal information can be found in [11: 80-86, 194-201] and [26].

We can conclude this sub-section by noticing that some simple *"auto-evolving" possibilities* of the HClass-like component (plain/static knowledge) of the GENIUS' "model of the world" are foreseen in the project. Apart from assuring a certain degree of flexibility and generality to the ontological tools – and of providing an opportunity of improving the model learning from experience – these possibilities will allow us to deal with a very practical problem. This comes up when the identification of an incoming GWE as an instance of some very specific concept of the HClass hierarchy is not possible – even in an approximate way, see squared_object instead of squared_table or object_grasping instead of bottle_grasping – because this last concept (and the associate specific/generic entities) is not (yet) present in the hierarchy. In this case, it is in general suitable (because of the obvious associated 'fuzziness' problems) to avoid to systematically associating the unknown GWE, as an instance, to the "root_" of the ontology (or some other very high-level concepts). The GENIUS' auto-evolving procedures are then based, presently, on a sort of "symbolic" approach where W3C reasoners in the style of RACER, Pellet, Fact++ or Hoolet [27] are used to insert *in the best way*, using then a "*subsumption*" approach, the 'provisional' concept (and its associated 'provisional' instance) represented by the unknown GWE with its URI-like identifier within the structures of the ontological hierarchy. Final operations concerning the attribution of a 'name' to the new concept and its instances – and the introduction, in case, of appropriate generic concept(s) in the hierarchy – are executed off-line making use of free available knowledge bases like, e.g., Roget's Thesaurus and Wordnet. Bayesian techniques for uncertainty management/learning could also be used in this context.

3.3 Full Recognition and Categorization of the GWEs

The conceptual 'representation of the world' mentioned above – a general ontology, including structures for representing both "simple/static" concepts and "structured events/situations" – is used in association to a set of *inference rules* to "recognise/categorize" the GWEs.

"Recognizing/categorizing" means to establish a *correspondence* between the known/unknown entities (objects, events, relationships, situations, circumstances etc.) coming from the external world and the high level conceptual and ontological representations proper to the "description of the world" – incrementing these last as experiences are accumulated. Implementing a full recognition and categorization of GWEs is a quite difficult task given that there is here no guaranty in general that the information available in the external environment could be sufficient to perform the recognition task in an adequately complete way – and this independently from the level of 'completeness' of the ontology. For instance, sensing (and inferencing) might not be capable of fully categorizing a table as such, leading then to the instance SQUARE_OBJECT_1 for an incoming GWE instead of the correct TABLE_1 or TOFFEE_BOX_1 instances. The GENIUS' world model must thus be sufficiently powerful to integrate with its own stored knowledge, in case, the lack of information associated with the provisional descriptions of the GWEs. Moreover, as stated at the end of 3.2 above, it must be also be sufficiently flexible to allow for the creation of new 'conceptual entities' in case of impossibility of associating a 'brute' GWE description with an existing one.

The "recognition/categorization" activities are performed, substantially, in *two subsequent phases*. In the first, the 'brute' descriptions of the GWEs directly derived from the outputs of the sensors included in the environment are linked to the corresponding conceptual entities included in the "plain/static" (standard) component of the world model introduced above. This is executed thanks, mainly, to the use of a semantic-based reasoning system able to *unify*, to the best possible extent, *the low-level features (properties/attributes) attached initially to the GWEs with the semantic properties of the general concepts included in this static component*. This conceptual unification activity could also be integrated with the usual, 'algorithmic' machine learning techniques used to recognize an object making use of a comparison of the associated features with those of 'standard' objects stored in a database (see methodologies like SIFT, SURF, David Lowe's method, etc.). When a GWE is recognized with a reasonable probability degree, a *new instance* of the corresponding concept is added to the corresponding 'branch' of the ontological world model.

Recognizing the GWEs represented by contexts, events, situations, circumstances – i.e., identifying the corresponding conceptual entities described in the "structured dynamic" components of the world model – consists, mainly, in a *reasoning process* based on the results of the previous process of recognition of the "physical/static" GWEs. Let us suppose, in fact, that a simple pattern recognition system could be used to identify the presence of a high-level GWE of the Move:AutonomousPersonDisplacement type. Before being able to add in the dynamic component of the world model a new instance of an event of this type, we must i) identify the possible instances that are candidate to fill the SUBJECT, OBJECT, SOURCE, MODALITY etc. properties (functional roles associated with the MOVE predicate) of this event; ii) verify that these entities satisfy the constraints associated with the above properties/roles; iii) verify the global coherence of the new instance with respect to the global situation we are dealing with. A pure 'cognitive-driven' approach cannot be sufficient, in the most complex cases, to completely solve the 'correspondence' problem. In a GENIUS context we are then evaluating the possibility of associating standard 'algorithmic' solutions – based, e.g., on temporal constraint propagation principle [28] – with the 'cognitive' ones.

3.4 Reasoning Based on the Full Recognition of an Existing Situation

When all the GWEs (objects, agents, events, situations, circumstances, complex events and scenarios, behaviours etc.) have been recognised, we make use of the general world description enriched with the corresponding instances to *take decisions* (such as event processing to choose which goal to pursue, or to change the current plan) and allow, in case, physical actions (like opening/closing gates/doors, allowing/disallowing switches etc.) on the original GWEs and their context.

We can see these reasoning activities as the implementation of a set of '*services*' – represented, mainly, by inference procedures based on the NKRL system of generalised "if/then rules" [29] – that are general enough to be used (in case, slightly adapted) in a vast range of GWEs applications and that are permanently stored on the GENIUS platform. 'Reasoning' ranges in NKRL from the *rich direct questioning* – using specific data structures called "search patterns" – of knowledge bases of structured/dynamic knowledge (high-level GWEs corresponding to descriptions of human and human-like behaviours, situations, simple and complex events, etc.) represented according to the NKRL format, to *high-level inference procedures*. These make use of the richness of the knowledge representation system to automatically establish *'interesting' relationships* among the different sorts of information. For example, the "*transformation rules*" employ a sort of analogical reasoning to try to automatically replace some queries that failed with one or more different queries that are not strictly 'equivalent' but only 'semantically close' to the original ones, disclosing then semantic affinities between what was originally requested and what is really present in the GWEs knowledge base. The "*hypothesis rules*" allow us to build up causal-like explications of given events/situations/behaviours according to pre-defined reasoning schemata formed of several reasoning steps; during the execution, these last are automatically converted by an inference engine into search patterns that try to find an unification with the contents of the knowledge base. Integrating these two inference modes lets us, among other things, to augment the possibility of discovering large amounts of the implicit information hidden within the general knowledge collected about the situation at hand.

Some possible examples of GENIUS-based services are detailed below; note that a GENIUS approach is also compatible with the majority of the IoT 'standard' applications like those listed, e.g., in [4: 2793-2797, 5: 11-19]:

- **Accident Avoidance.** This type of inference refers, e.g., to situations like that schematically illustrated in the example of Fig. 1, where the goal consists in preventing a dependable person with vision troubles (or a robot or a baby) to collide with potentially dangerous objects. This activity must be based on an in-depth knowledge of the (a priori unknown) GWEs in the environment to let the system decide whether a given object should be absolutely avoided (i.e., a table) or it can in principle be stepped on (a newspaper, a magazine, a puddle…). The same type of inference can be easily generalized in a context of homeland security, of driving control, of exploration of unknown territories by a rover, of butler robots, etc.

- **Planning.** This set of reasoning activities could concern the optimisation, e.g., of the surveillance tasks of a dependable person, or the creation of an adaptive 'buying path' within a supermarket, or the more mundane task of preparing a cold drink by going to the cupboard to get a cup, going then to the fridge, opening the fridge and taking out the juice and eventually pouring the juice into the cup. Planning includes prioritising the goals, establishing when goals are complete, determining when the system is required to re-plan, etc. Once again, this sort of activity is proper to a very wide set of GWEs applications.
- **Monitoring**. Monitoring concerns a large class of possible applications, from those related to the control of an elderly person in homecare after hospitalisation (see also the scenario in the following Section) – when the system detects a fall down from stairs, it must order hospital emergency intervention and move a house care mobile robot, embedded with a camera, to connect the (supposedly still conscious) elderly with the hospital, using the camera and vital signal sensors to evaluate her/his health state while waiting the emergency staff arrival – to the prevention of terrorism activities (bomb disposal etc.), decontamination of lands and buildings, building security, identity management, gas/oil plants inspection and supervision, etc. In all these cases, different independent GWEs of a diverse degree of complexity must be identified/categorized, and then aggregated/correlated to represent complex events/situations.
- **Intentions/Behaviours Detection.** Inferences of this type have normally (but not necessarily) GWEs of the human type as central characters. They can be associated with "monitoring" activities when, in the "elderly monitoring" situation evoked above, it is necessary to infer from her/his itinerary and actions that the old person manages to get away from the house instead of going into the kitchen/toilet, or when the unfriendly intentions of an intruder must be noticed. But they also concern a wide range of "sociological" applications as detecting particular behaviours in young people denoting a possible pathway to social rejection, inferring attitudes towards "health related" fields like the adoption of condoms, leisure, exercise or diet, carrying out perspectives studies concerning the behaviour of shoppers or intentions of (human or automatic) drivers, etc. The use of NKRL inference techniques of the "hypothesis" type is particularly appropriate in all these cases.

4 An "Assisted Living" Real-Life Scenario

As well known, dealing with the problems concerning the monitoring of elderly people at home is becoming particularly urgent. Today, one in three people over

80 years in Europe wishes to retain some form of independence as far as possible and continues then to stay at home. These aged people often experience some form of cognitive impairment: memory problems and mild dementia are widespread among the European 65+ population (9.9 million people are estimated to suffer from different types of dementia in Europe). They must then be assisted in tasks like: preparing meals, bathing, dressing, feeding, grooming, getting around, etc. Moreover, they must also be monitored to prevent any possible safety hazards including: high risk of falls, fire hazards, improper use of sharp objects, medicines, hazardous household products, water temperature, disorientation and wandering and therefore risk of getting lost, inadequate drinking and eating, etc. See, e.g., [30] for a collection of recent papers dealing with elderly homecare and connected topics in a generic IoT context.

In a GENIUS context, we are exploring how the GWE paradigm can be of help to handle the elderly at home problem. A GWE-based application for the monitoring of elderly people is then actually developed, see Fig. 2 for an overall architectural scheme. The application is concretely structured into a set of real-life "*Ambient Assisted Living (AAL)*" scenarios like:

- It is a common practice not to supervise an elderly person with mild dementia during the night: the caregiver accompanies her/him to bed and close the bed edges to prevent falls. Unfortunately, disoriented persons might try to climb over the edges, exposing themselves to high risks of fall. A sensor can detect the attempt to get down from the bed, allowing then the monitoring system to send an alert to the caregiver. Should the person fall before the arrival of the caregiver, the monitoring system can detect the fall and call the emergency services.
- Dementia and cognitive impairments affects memory. It is very likely that the elderly person forgets taking her/his pills at due time and/or takes them more times than needed. The monitoring system can check the assumption, remembering which pills should be taken and when and asking the elderly person to confirm that they have been taken. The system could then refuse to supply those pills that have already been taken and/or alert the caregiver in case of misuse.
- A cognitive impaired elderly person can put her/himself at risk making inappropriate use of household equipment, such as the stove or the boiler. Such behaviours can be detected by the monitoring system that, for precaution's sake, could stop, e.g., the gas provision. The caregiver would then be notified and – if considered safe – the command of turn the gas on again could be provided.

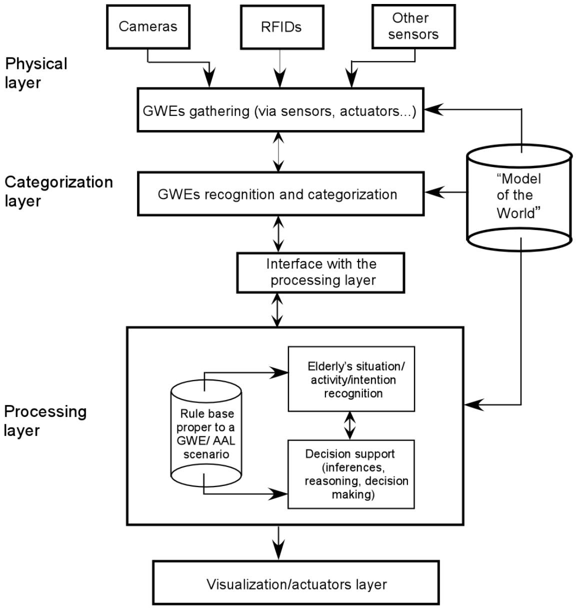

Cameras
RFIDs
Other sensors
Physical layer
GWEs gathering (via sensors, actuators...)
Categorization layer
GWEs recognition and categorization
"Model of the World"
Interface with the processing layer
Processing layer
Elderly's situation/ activity/intention recognition
Rule base proper to a GWE/ AAL scenario
Decision support (inferences, reasoning, decision making)
Visualization/actuators layer

Fig. 2 Overall architecture of a GWE-based AAL application

The first AAL scenario mentioned above, when defined in some details, can then correspond to the structured description below. For its practical implementation, we have adopted the solution of using a *mobile robot* to realize the interaction between the monitored elderly person ("Mary") and the monitoring system: this sort of solution offers several advantages. First of all, the interaction between the system and the person can be achieved in a more 'personal style' than anonymously in the form of 'voice from the off'. The robot could serve for bidirectional interaction with a remote caregiver by monitor/camera loudspeaker/microphone; it should also permit remote control by caregiver for inspection or to start communication. At the same time, the robot could be able to act autonomously, e.g., for keeping contact with the person, for giving advices etc. For that, it must be able to identify situations, intentions, motions of the person. This is to be done in close cooperation with fixed sensors of different kinds for supervision that are placed at the appropriate locations (e.g., sensors on the ground to detect a falling down, heat sensors …). A simple Pekee II Mobile robot – already mentioned in sub-section 2.2 above – is concretely used in the experiment. The scenario's steps are detailed in the following; some of the *high-level GWEs (in NKRL format) generated by the monitoring process* are detailed for exemplification's sake in Table 1:

- **At t_0**, Mary's caregiver leaves the house, after having put Mary in bed. The monitoring system recognizes the related GWEs from the pressure sensors in the bed, from the cameras (fixed ones, robot camera), and from the wearable sensors of Mary's clothes observing their health state like pulse rate, breathing rate, body temperature and their motions by gyro and accelerometer. According to 3.1 above, the data are collected and interpreted at the lower level to identify the GWEs representing the physical environment and to infer the corresponding conceptual GWEs describing the situation and the occurring event. Making use of the world knowledge (3.2), the scene is identified as "Mary starts to sleep in her bed" (aal1.gwe8 of Table 1). No actions are needed.

- **At t_0+120**, many new GWEs (physical, conceptual) are detected (3.3), which do not correspond with the expected event "Mary is sleeping" – see also the "processing layer" in Fig. 2. At the physical level, the pressure sensors in the bed measure forces that are not consistent with a sleeping person. Mary's wearable sensors measure high activity and the gyro/acceleration sensors can be interpreted as an attempt to climb over the bed edges. Note that it is not necessary to have a unique description of sensor measurements for the climbing situation, but the combination of all information using GWEs on several interpretation layers gives enough evidence for the conclusion by the rule system (3.4), that Mary has the intention to leave her bed (aal1.gwe27 of Table 1). Motion capturing (3.1) can give further evidence. The monitoring system therefore activates an alert of level 1: the lights in the room are turned on to give Mary a better understanding of the situation. A vocal message (uttered by the robot) says to Mary to stay in bed and that someone is coming to help her. For security's reasons, the robot keeps a safe distance to the bed. He is able to do that because he has the necessary world knowledge (3.2) and can plan appropriate movements by inference and planning (3.4).
- **At t+121**, the monitoring system gets further information from the cameras and sensors. The data still coincides with the event "Mary leaves her bed". It can be inferred, from the capturing of motion data (3.1), that she is even making progress in her attempts. The system triggers a level 2 alert: according to the action rules (3.4) a phone call is sent to Mary's caregiver, informing her/him of the situation.
- **At t+122**, the system detects that the bed is empty: There are no more GWEs (from pressure data of the bed) related to the situation "Mary is inside the bed", but pressure sensors on the ground nearby the bed detect new forces that are translated to conceptual, high-level structured GWEs describing the new situation. The motion data from the wearable clothes as well as the interpretation of the camera images confirm this conclusion by additional conceptual GWEs (3.3). Hence the monitoring system has to take measures (3.4) to take care of Mary's health. Through the sensors monitoring her health state, the system can conclude that she is still alive, but it is not clear if she can still apprehend her environment and her situation. The next action according to the predefined rules of behaviour (3.4) is then to check Mary's state: the system (the robot) asks Mary to press a button to confirm she is fine (aal1.gwe47 and aal1.gwe48 of Table 1).
- **At t+124**, the system does not register any active response by Mary: there is no clear voice signal, and no intentional motion can be detected by cameras or

Table 1 Examples of high-level GWEs corresponding to the first AAL scenario

```
aal1.gwe8)     BEHAVE     SUBJ       MARY_: (BEDROOM_1)
                          MODAL      sleeping_
                          { begin }
                          date-1:    t0
                          date-2:
Behave:HumanProperty (1.1)
```

At time t_0, Mary begins to sleep.

Comment: This high-level GWE is an instance of the "template" (see 3.2 above) Behave:HumanProperty, used to denote general situations involving human beings. BEHAVE is a conceptual predicate, SUBJ(ect) and MODAL(ity) are functional roles, MARY_ and sleeping_ are the arguments of the predicate introduced through the functional roles. MARY_ is associated with a determiner of the "location" type, BEDROOM_1, through the ":" operator. MARY_ and BEDROOM_1 are instances of HClass (3.2) concepts that designate low-level GWEs already recognized and categorized; sleeping_ is simply an HClass concept pertaining to the animate_activity branch. begin is a "temporal modulator", used to denote that the date inserted in the date-1 block corresponds to the commencement of the state represented by the global high-level GWE.

```
aal1.gwe27)    MOVE       SUBJ       MARY_: (BED_1)
                          OBJ        MARY_: (BEDROOM_1)
                          { wish }
                          date-1:    t0+120
                          date-2:
Move:AutonomousPhysicalPersonDisplacement (4.311)
```

At time t_0+120, Mary plans to move from the bed to the bedroom.

Comment: The template Move:Autonomous… is characterized by the following properties: i) the entity that is moving is considered as moving itself as an OBJ(ect); ii) the initial location is associated with the filler of the SUBJ(ect) role, whilst the arrival location is associated with the filler of OBJ(ect). wish is a "modal modulator" that, as all the operators of the "modulator" type, takes the whole GWE as its argument.

```
aal1.gwe47)    MOVE       SUBJ          ROBOT_1
                          OBJ           #aal1.gwe48
                          BENF          MARY_
                          MODAL         vocal_message
                          CONTEXT       (SPECIF control_ (SPECIF physical_state MARY_))
                          date-1:       t0+120
                          date-2:
Move:StructuredInformation (4.42)
```

At time t_0+120, the robot sends to Mary a vocal message, whose content is described in aal1.gwe48.

Comment: This complex GWE illustrates what is called "completive construction" in NKRL, one of the simplest means to deal with those "connectivity phenomena" introduced above in 3.2. This construction is used, among other things, to represent any sort of transmission of plain messages, commands etc. The 'content' of the message is then denoted by another, independent GWE (aal1.gwe48 in our case) whose conceptual label is used as 'filler' of the OBJ(ect) functional role, whilst the recipient of the message (MARY_) is the filler of the BEN(e)F(iciary) role. With respect to the filler of the CONTEXT role, this represents an example of "expansion" (structured predicate argument), where the SPECIF(ication) operator is used twice to signify that the "control" is about a given "physical state" and that this physical state is proper to "Mary".

```
aal1.gwe48)    PRODUCE    SUBJ          MARY_
                          OBJ           button_pushing
                          TOPIC         LIFE_SAVING_BUTTON_1
                          { oblig }
                          date-1:       t0+120
                          date-2:
Produce:PerformTask/Activity (6.3)
```

At time t_0+120, Mary must, modulator oblig(ation), *press the* LIFE_SAVING_BUTTON_1.

Comment: oblig, fac(ulty), interd(iction) and perm(ission) are the four NKRL "deontic" modulators.

gyro/acceleration sensors in Mary's clothes. Note that this is not a trivial task, because unconscious motions can still occur. Again, a combination of different physical signals over some time (3.1) and a layered interpretation leading to related conceptual GWEs (3.3) is necessary to get enough evidence that Mary needs in fact human help (inferences, 3.4).

- **At t+125**, then, an alert is sent to the emergency services. A visual communication is established and the emergency services doctor sees Mary lying on the floor (by the fixed cameras and the robot camera). She/he can move the robot from remote to get closer information. She/he also gets the measured data of Mary's health state. According to that information, the doctor decides to ask for an ambulance.
- **At t+126**, a vocal message tries to inform Mary that rescues are arriving and a phone call informs the caregiver that an ambulance is on its way.
- **At t+130**, the system detects a presence in front of the door and opens this door.
- **At t+131**, the emergency services arrive.

5 Conclusion

In this Chapter, we have supplied a short description of an on-going project, GENIUS, which takes its inspiration from an advanced interpretation of the IoT's aims where the possibility of interpreting the *environmental and context information*, of detecting information related to *human intentions/behaviours*, of enabling *human-like inferences and multi-modal interactions*, and eventually of *acting on behalf of the users' intentions* are particularly important. Among the novelty aspects introduced by the project, we can remark:

- The development of *methods and formalisms* for describing, recognizing and reasoning about the occurrences of "*Generalized World Entities*" (GWEs). The key property of GWEs concerns the fact that they are not limited to "*physical objects*", see the discussion in Section 1 about the definition of the domain of interest covered by the "IoT" term. This is the case, on the contrary, in most current state of the art approaches to anchoring and situation/activity recognition. Rather, the GWE paradigm provides a *uniform formalism* (a uniform context) to represent (known and unknown) objects, agents, events, situations, circumstances behaviours etc. and their evolution in time, as well as the relationships between all these entities.
- The development of a *general framework (and of the related practical tools)* for establishing and maintaining the links between *the conceptual GWE descriptions and their instances* as they are perceived – at different levels of complexity, e.g., physical objects and structured events – from the real world through *sensors*. These conceptual descriptions correspond to the entities included in a broad-spectrum ontology, auto-evolving and capable of describing both the "static" and "dynamic" characteristics of the GWEs, and then to generalize starting from observed situations. Note that the possibility of creating an easy passage from the "*sensor level*" to the corresponding

"*conceptual level*" is crucial for adding 'intelligence' to the present IoT paradigm.

- The development of *reasoning (inference) procedures*, based on the *full recognition* of all the GWEs (objects, agents, events, complex events, situations, circumstances, behaviours etc.) involved in a given application scenario and implemented under the form of a set of "*services*". These services must be general enough to be used (with slight adaptation) in a wide range of GWE-based applications. Possible examples of these services, of a different degree of complexity and generality, are wide-ranging procedures for dealing with collision avoidance, monitoring or planning.

The project foresees also the implementation of a *flexible and extensible architecture* able to support any sort of 'intelligent' applications based on the GWE paradigm. The platform should then be 'open' enough to allow the easy plug-in of i) new general modules that could be needed in order to improve the functioning of the platform; ii) specific GWE applications in particular fields. Details about the use of the GWE paradigm to set up an "Ambient Assisted Living (AAL\i)" application for dealing with the "elderly at home problem" are provided in the Chapter.

Acknowledgments. The author is indebted to Prof. Dr. Hans-Dieter Burkhard, Computer Science Department of the Humboldt-Universität Berlin, and to Dr. Licia Boccaletti, Co-operative "Anziani e non solo" ("Not only elderly people") in Carpi (Italy), for their valuable help in establishing the final version of the AAL scenario detailed in Section 4 above. He thanks also the editors of the "Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence" book and the anonymous referees for their comments and suggestions, which have significantly contributed to improve the quality of this chapter.

References

1. National Intelligence Council: Disruptive Civil Technologies, Six Technologies With Potential Impacts on US Interests Out to 2025 (Conference Report CR 2008.07. NIS, Washington, DC (April 2008), http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA519715
2. Santucci, G.: The Internet of Things, Between the Revolution of the Internet and the Metamorphosis of Objects. Brussels: CORDIS Publications and Reports, Internet of Things and Future Internet Enterprise Systems, Brussels (2010), http://cordis.europa.eu/fp7/ict/enet/documents/publications/iot-between-the-internet-revolution.pdf
3. Uckelmann, D., Harrison, M., Michahelles, F.: An Architectural Approach Towards the Future Internet of Things. In: Uckelmann, D., Harrison, M., Michahelles, F. (eds.) Architecturing the Internet of Things, pp. 1–24. Springer, Heidelberg (2011)
4. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A Survey. Computer Networks, The International Journal of Computer and Telecommunications Networking 54, 2787–2805 (2010)

5. Cluster of European Research Projects on the Internet of Things (CERP-IoT): Internet of Things Strategic Research Roadmap – 15 September, 2009. European Commission DG INFSO-D4 Unit, Brussels (2009), http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf
6. Bucherer, E., Uckelmann, D.: Business Models for the Internet of Things. In: Uckelmann, D., Harrison, M., Michahelles, F. (eds.) Architecturing the Internet of Things, pp. 253–277. Springer, Heidelberg (2011)
7. Hribernik, K.A., Hans, C., Kramer, C., Thoben, K.-D.: A Service-oriented, Semantic Approach to Data Integration for an Internet of Things Supporting Autonomous Cooperating Logistics Processes. In: Uckelmann, D., Harrison, M., Michahelles, F. (eds.) Architecturing the Internet of Things, pp. 131–158. Springer, Heidelberg (2011)
8. Capezio, F., Mastrogiovanni, F., Sgorbissa, A., Zaccaria, R.: Towards a Cognitive Architecture for Mobile Robots in Intelligent Buildings. In: Proceedings of the ICRA 2007 Workshop on Semantic Information in Robotics, pp. 13–20. IEEEXplore, Piscataway (2007)
9. Martinez Mozos, O., Jensfelt, P., Zender, H., Kruijff, G.-J.M., Burgard, W.: From Labels to Semantics: An Integrated System for Conceptual Spatial Representations of Indoor Environments for Mobile Robots. In: Proceedings of the ICRA 2007 Workshop on Semantic Information in Robotics, pp. 25–32. IEEEXplore, Piscataway (2007)
10. Saffiotti, A., Broxvall, M., Gritti, M., LeBlanc, K., Lundh, R., Rashid, J., Seo, B.S., Cho, Y.J.: The PEIS-Ecology Project: Vision and Results. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), pp. 2329–2335. IEEEXplore, Piscataway (2008)
11. Zarri, G.P.: Representation and Management of Narrative Information – Theoretical Principles and Implementation. Springer, London (2009)
12. Zarri, G.P.: Knowledge Representation and Inference Techniques to Improve the Management of Gas and Oil Facilities. Knowledge-Based Systems (KNOSYS) 24, 989–1003 (2011)
13. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-Time Human Pose Recognition in Parts from Single Depth Images. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1297–1304. IEEEXplore, Piscataway (2011)
14. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57, 1371–1354 (2004)
15. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5, 199–220 (1993)
16. Novalija, I., Vassileiou, H., Jermol, M., Bradeško, L.: Formalized EURIDICE Domain Knowledge – Deliverable D12.2 (Final Draft, version 2.0). EC EURIDICE Integrated Project – ICT 2007-216271 (2009), http://www.euridice-project.eu/euridice_rep_new/files/PublicDocs/pub/Public%20Deliverables/EURIDICE_D122_Formalized_Domain_Knowledge_V2.0.pdf
17. Velasco, J.R., Vega, M. (eds.): Contributors: SotA Report, Smart Space DIY Application Creation and Interaction Design – Deliverable D4.1 (Final Version, 3.0). Do-it-Yourself Smart Experiences (DiYSE) ITEA 2 project 08005, http://dyse.org:8080/download/attachments/4816946/DiYSE_D4.1_SotA+report+Smart+Space+DIY+application+creation+and+interaction+design_RELEASED.pdf?version=1

18. Zarri, G.P.: Differentiating Between "Functional" and "Semantic" Roles in a High-Level Conceptual Data Modeling Language. In: Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference, FLAIRS-24, pp. 75–80. AAAI Press, Menlo Park (2011)
19. Noy, N.F., Fergerson, R.W., Musen, M.A.: The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 17–32. Springer, Heidelberg (2000)
20. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A. (eds.): OWL Web Ontology Language Reference, W3C Recommendation. W3C (February 10, 2004), http://www.w3.org/TR/owl-ref/
21. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language Primer, W3C Recommendation. W3C (October 27, 2009), http://www.w3.org/TR/owl2-primer/
22. http://www.schema.org/docs/documents.html
23. Mizoguchi, R., Sunagawa, E., Kozaki, K., Kitamura, Y.: A Model of Roles Within an Ontology Development Tool: Hozo. Journal of Applied Ontology 2, 1591–1579 (2007)
24. Salguero, A.G., Delgado, C., Araque, F.: Easing the Definition of N–Ary Relations for Supporting Spatio–Temporal Models in OWL. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 271–278. Springer, Heidelberg (2009)
25. Liu, W., Liu, Z., Fu, J., Hu, R., Zhong, Z.: Extending OWL for Modeling Event-oriented Ontology. In: Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 581–586. IEEE Computer Society Press, Los Alamitos (2010)
26. Zarri, G.P.: Representation of Temporal Knowledge in Events: The Formalism, and Its Potential for Legal Narratives. Information & Communications Technology Law 7, 213–241 (1998)
27. http://www.w3.org/2007/OWL/wiki/Implementations
28. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier, Amsterdam (2006)
29. Zarri, G.P.: Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 304–340. Springer, Heidelberg (2005)
30. Vasilakos, A.V., Chen, H.-H., Mouftahm, H., Habib, I., Montgomery, K., guest editors: Special Issue on Wireless and Pervasive Communications for Healthcare. IEEE Journal on Selected Areas in Communications 27(4), 361–574 (2009)

Plugging Text Processing and Mining in a Cloud Computing Framework

Akil Rajdho and Marenglen Biba

Abstract. Computational methods have evolved over the years giving developers and researchers more sophisticated and faster ways to solve hard data processing tasks. However, with new data collecting and storage technologies, the amount of gathered data increases everyday making the analysis of it a more and more complex task. One of the main forms of storing data is plain unstructured text and one of the most common ways of analyzing this kind of data is through Text Mining. Text Mining is similar to other types of data mining but the problem is that differently from other forms of data that are properly structured (such as XML) in text mining data in the best case scenario is semi-structured. In order for them to derive valuable information, text mining systems have to execute a lot of complex natural language processing algorithms. In this chapter we focus on text processing tools dealing with stemming algorithms. Stemming is the step that deals with finding the stem (or root) of the word which is essential in every text processing procedure. Stemming algorithms are complex and require high computational effort. In this chapter we present an Apache Mahout plugin for a stemming algorithm making possible to execute the algorithm in a cloud computing environment. We investigate the performance of the algorithm in the cloud and show that the new approach significantly reduces the execution time of the original algorithm over a large dataset of text documents.

1 Introduction

Computational approaches are rapidly evolving in the last years leading to novel and faster architectures that have the potential to boost computing throughput. Many years ago the most used computing approach for complex problems and tasks was Distributed Computing which consists in solving a problem by using a number of independent computers connected in a network. After that Parallel Computing emerged trying to solve the problem by dividing it into subproblems that were executed simultaneously. Later Grid Computing which involved the combination of computer resources from different geographical or administrative

Akil Rajdho · Marenglen Biba
Department of Computer Science, University of New York in Tirana, Tirana, Albania
e-mail: akilrajdho@gmail.com, marenglenbiba@unyt.edu.al

Akil Rajdho
School of Computing and Mathematical Sciences, University of Greenwich, London, UK
e-mail: akilrajdho@gmail.com

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 369–390.
DOI: 10.1007/978-3-642-34952-2_15

locations to solve a task, was introduced and nowadays researchers and developers are switching to Cloud Computing which can be defined as a type of distributed computing paradigm augmented with a business model via a Service Level Agreement between providers and consumers [10].

With new data collecting and storage technologies, the amount of gathered data is rapidly increasing everyday making the analysis of it a more and more complex task. Data is stored in different locations and in different forms. But raw data is almost useless without proper analysis and research done upon it. One of the main forms of storing data is plain unstructured text and one of the most common ways of analyzing this kind of data is through Text Mining. Text Mining is similar to other types of data mining but the problem is that differently from other forms of data that are properly structured (such as XML), in text mining data in the best case scenario is semi-structured. In other words, Text Mining which can be considered part of Artificial Intelligence is the process of extracting knowledge from text. Another growing research area is Information Retrieval (IR) which deals with finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections usually stored on computers [16]. In order to derive valuable information IR systems have to execute a lot of complex algorithms that mostly regard language processing tasks. In this chapter we focus on text processing tools dealing with the stemming task which is finding the stem (or the root of the word). This step is essential in every text processing procedure. As part of Natural Language Processing (NLP) these kind of algorithms are very complex and require high computational capacity that does not come neither for free nor cheap.

A growing number of large Internet of Things (IoT) systems and applications would feed text data to cloud, needing processing and mining. An example of this kind of data is the text coming from online social networks. Millions of text comments are generated every day in online social platforms and understanding these for viral marketing has an outstanding business value. However, the amount of data is such that no single processing service can handle it, therefore these online systems may feed their text comments to the cloud. In some cases, an immediate processing may also be very useful. For example, in analyzing user comments on products, sentiment analysis or opinion mining may help online systems propose a certain product to a user based on her comments on another product. In most cases, a collective inference process based on many product reviews may help the producing or selling company revise or better customize the product. This is just an example of the potentials of exploiting the cloud for text processing in order to boost online business. Other examples for exploiting the cloud in a IoT context include: news filtering systems that automatically classify text coming from multiple sources; relationship extraction systems that find useful relation patterns in the text; named entity recognition systems for the identification of relevant entities in a certain text; machine translation systems that make possible the dynamic generation of translated text in the required language; word sense disambiguation systems that make possible the identification of the right sense of a certain word, etc. All these systems require usually very large quantities of text to be trained and have high computational demand which makes the cloud the perfect solution to the requirements for this kind of systems.

In addition, large online systems usually do not provide services for languages with small scope, and the possibility of delegating some of these services to the cloud for this kind of languages may be possible with the approach proposed in this chapter. One of these services is machine translation in rare languages or languages of small nations. This kind of service may have small return-of-investment for the large company but may have a large local impact on the small country. Providing text processing capabilities in the cloud which is fed with raw text from online systems, may definitively lead to outstanding improvements of IoT systems experience in small countries. A significant example is the localization of online social platforms which usually does not come under the local language due to the difficulties in customizing such platforms. With the evolution of more approaches like ours, the full potential of the cloud may lead to a global network of online systems that easily change their face from country to country using language and text services from the cloud.

A stemming algorithm is an algorithm that is concerned with suffix and prefix removal in words. In this chapter we focus on stemming of Albanian which is a language not much investigated. In fact, the first stemming algorithm ever for the Albanian language was developed only in 2011 [24]. Here we present an Apache Hadoop plugin for the above stemming algorithm, using Apache Mahout, making therefore possible to execute the algorithm not only in a local machine but also in a cloud computing environment. Moreover the algorithm is investigated for possible improvements which will lower the execution time. The morphology analysis of the Albanian language is not reconsidered in this chapter. After the implementation of Cloud Computing infrastructure and the development of the plugin that runs in it, the same amount of data as in the experiments done in [24] is given as input to the plugin in order to compare execution time differences between local computing and Cloud Computing. After that a more stressful experiment with more documents is performed with both algorithms to evaluate and compare their performances. The Cloud environment will offer both Software as a Service (SaaS) through the plugin and Hardware as Service (HaaS) providing powerful software and hardware for the execution of the algorithm. The plugin will be publicly available to all those interested to further develop the stemming algorithms and natural language tools for the Albanian language.

The chapter is organized as follows: Section 2 introduces cloud computing as a computing paradigm, Section 3 presents information retrieval and text mining, focusing on the language processing steps and in particular in the stemming stem. Section 4 describes the Apache Mahout framework, Section 5 describes the development of the plugin, Section 6 presents the experimental evaluation and we conclude in Section 7.

2 Cloud Computing

As more facets of work and personal life move online and the Internet becomes a platform for virtual human society, a new paradigm of large-scale distributed computing has emerged. Web-based companies, such as Google and Amazon, have built web infrastructure to deal with the internet-scale data storage and computation [21]. This means that a lot of researchers can benefit from such emerging infrastructures in order to conduct their experiments faster and cheaper.

In such an evolving information world, most companies and research centers are looking for new ways and resources to handle their needs, those of their employees, clients and products. The rapid change of technology has brought into existence Cloud Computing. Cloud Computing is a recent trend in Information Technology (IT) that moves computing and data away from desktop and portable PCs into large data centers. It refers to applications delivered as services over the Internet as well as to the actual cloud infrastructure — namely, the hardware and systems software in data centers that provide these services [7]. Because powerful computers are so expensive and hardly ever people need them permanently, it has become more feasible to use Cloud Computing for large and complex computational tasks. Cloud Computing is becoming so popular because just like web applications it only requires internet connection and it doesn't matter how powerful the computer is in the client side. Among other advantages, Cloud Computing fundamentally changes the way that IT services are delivered to organizations. Instead of both owning and managing IT services for themselves, or using an outsourcing approach built around dedicated hardware, software, and support services, organizations can use cloud computing to meet their IT requirements using a flexible, on-demand, and rapidly scalable model that requires neither ownership on their part, nor provision of dedicated resources [2].

2.1 Service Delivery Models

Cloud Computing delivers services to the customers in different ways according to their needs. Among the variety of Cloud Computing service delivery models the most popular are Software as a Service, Platform as a Service and Infrastructure as a Service also known as Hardware as a Service. These three more important service delivery models are discussed in more details below.

2.1.1 Software as a Service (SaaS)

SaaS also known as Software on Demand is the kind of service offered by the cloud where software is rented from a service provider rather than being bought by individuals or companies. Instead of buying licenses the customer is charged on a pay per use paradigm. All other costs like maintenance, backups, optimization and load balancing are taken care of by the service provider. So the customer cuts out maintenance and license costs. SaaS also enables software companies to prevent unauthorized distribution of software by users. The only problem is that most of the times the customer has limited access to software customization according to his needs because a certain application in the cloud can be shared among a variety of users. Some popular SaaS service providers that also offer some services free of charge are Google (google docs), Microsoft (Microsoft Online Services).

2.1.2 Platform as a Service (PaaS)

Service providers offer to the customer access to platforms so they can deploy their applications there. They do not have any control on the network or on the hardware deployment methods. The technical framework is provided to the customer without him spending his time on thinking and planning hardware requirements, installing operating systems and load balancing. When the number of users for web application grows, PaaS is the optimal solution because the platform can be ready in a matter of minutes or maximum hours making the developers focusing on the development of the application rather than taking care of prices of hardware and licenses of operating systems and it provides a faster time to market an application developed. Some of the popular PaaS providers are Google (Google App Engine) and Microsoft (Microsoft Azure).

2.1.3 Infrastructure as a Service (IaaS)

IaaS, the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications [18]. The term IaaS in other words means to rent a big infrastructure of servers or even an entire datacenter. It gives the possibility to customers to have a lot of control over the infrastructure, customizing it according to their particular needs. This kind of service enables businesses to build up very fast their projects and a lot cheaper than in traditional IT infrastructures.

2.2 Deployment Models

Deploying Cloud Computing can differ depending on requirements, and the following four deployment models have been identified, each with specific characteristics that support the needs of the services and users of the clouds in particular ways [6].

2.2.1 Private Clouds

In private clouds the cloud infrastructure may be hosted inside the company or by a third party but in both cases the infrastructure is exclusively available only for one company and the resources are not shared with other companies. The features and functionalities are driven by the business requirements of the company hosting or renting the Cloud. This kind of Cloud Computing is used by companies that want to have more control over their data and since the cloud is dedicated only to a single company, additional security can be enhanced. Despite being private, this kind of Cloud Computing still has the benefits that Cloud Computing offers in general. For example resources can be added and rescaled on demand. This means that differently form public clouds where you have to fit your business to meet the clouds requirements in private clouds you can do the contrary, that is, make the cloud fit your business. Having a private cloud does not necessarily mean that companies must have a team of experts running and maintaining it. A company can choose to host the cloud by another party and have them take care of every maintenance and changes that the business needs.

2.2.2 Community Clouds

Community Cloud is a kind of cloud computing infrastructure where the resources are shared among a number of organizations with similar interests [6]. A community cloud could be built for health care services, federal services etc. This cloud infrastructure has usually more users than private cloud computing and less users than public cloud computing, and because usually this kind of clouds are "sponsored" by government institutions there is a higher access security level than other cloud infrastructures. Maintenance of the infrastructure is done usually by a third party.

2.2.3 Public Clouds

The cloud infrastructure is available to the public on a commercial basis by a cloud service provider. This enables a consumer to develop and deploy a service in the cloud with very little financial outlay compared to the capital expenditure requirements normally associated with other deployment options [7]. Public clouds are the most spread ones because the companies that provide the infrastructure gain profit out of it. Among the most popular Public Clouds are Microsoft Azure, Amazon Elastic Compute Cloud and Google Apps. The popularity of the public clouds is increasing day by day because people and business want to focus mainly on the development part not worrying too much about the maintenance.

3 Information Retrieval and Text Mining

Information Retrieval (IR) together with Natural Language Processing (NLP) are two important subfields of Artificial Intelligence (AI). NLP is normally used to describe the function of software or hardware components in a computer system which analyze or synthesize spoken or written language [11] while IR is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within collections stored in large repositories [16]. Text mining on the other hand is concerned with deriving valuable information usually from unstructured text. NLP, IR and text mining have much in common to achieve a certain goal. One of those cases where contribution from all the areas is needed is in stemming which is the process of finding the root of words by removing suffixes and prefixes. This process is not straightforward and requires techniques that come from language processing areas but also other disciplines.

3.1 Information Retrieval

The most common usage of IR is ad hoc retrieval where the user expresses the information that he is looking for in form of a query to the system and then the system displays him a list of relevant documents. Ad hoc retrieval is widely used in search engines like Google, Yahoo or Bing. In this particular scenario a user enters some words of interest for example “Albanian Tourism” in the search engine and then the search engine comes back to the user and displays all the pages that contain the words Albanian Tourism. The list of the pages displayed to the user is obtained by several algorithms that analyze the contents of the web pages and then store information about those pages in a more structured way in large indexed databases. Other important tasks in which IR techniques are employed include support of the user in browsing or filtering document collections, text classification, text clustering, cross-language retrieval, and multimedia retrieval [16]. IR is similar to Data Retrieval (DR) with some slight differences. While in IR the user uses queries in natural language to obtain documents with similarity matching, in DR the user uses queries that accept only a set of predefined syntaxes or keywords to retrieve documents that have an exact match. In formulating and processing queries, an essential task is to have the root of the word which helps to better filter the results.

3.1.1 The Retrieval Process

Information Retrieval as a process goes through a set of stages. Most of the documents cannot be fitted into an IR system as they are but they have to go through a set of pre-processing steps such as indexing, removal of stop words and stemming before the IR system can parse and then index them. The user plays an important role in the entire IR system because she is the one in the need of information so she searches for information by entering a query usually consisting of keywords and then the system matches these keywords against the indexed documents and displays the results to the user.

3.1.1.1 Preparation of Documents

Real-world data may be incomplete, noisy, and inconsistent, which can disguise useful patterns. This is due to: Incomplete data, lacking attribute values, lacking certain attributes of interest, or containing only aggregate data; Noisy data, containing errors or outliers; Inconsistent data, containing discrepancies in codes or names [27]. Moreover a cleaning of the documents which involves removal of duplicates, removal of unnecessary parts and parts that will not be indexed as well as inconsistent document removal should be done.

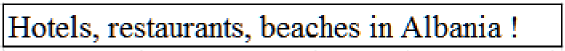
3.1.1.2 Document Pre-processing

After the preparation of documents is completed, individual documents are treated as a big stream of characters. Pre-processing also deals with removing useless tags remaining from type-setting information in newspaper archives, taking away “non-textual” items such as horizontal line and page-break tags in HTML documents, or in electronic mail smileys — :-) or :-(— and quotation representation (such as at the beginning of the line) and eliminating parts which do not belong to natural languages: mathematical or chemical formulae, programs**.** Conversely additional pre-processing is often associated to the segmenting into word units: acronym and abbreviation recognition, hyphenation checking, number standardization, etc [9].

3.1.1.3 Tokenization

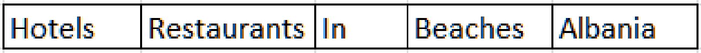
Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens*, perhaps at the same time throwing away certain characters, such as punctuation [16]. A simple example of tokenization would be,

Input string



Hotels, restaurants, beaches in Albania !

Output tokens:



Hotels	Restaurants	In	Beaches	Albania

As it can be observed after tokenization the commas as well as the "!" were removed and 5 individual tokens were created.

A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A *type* is the class of all tokens containing the same character sequence. A *term* is a (perhaps normalized) type that is included in the IR system's dictionary [16]. There are several ways to decide how and where to chop tokens, alternating from splitting at white spaces and removal of punctuation to more advanced techniques where hyphening, numbers and other rules varying from the language the query is submitted are taken in consideration. If the wrong method of tokenization is used the information returned to the user can be wrong. For example in English and many other languages hyphen sign (-) can be used in the middle of one single word like co-operation and with a normal tokenization we would end up with two tokens "co" and "operation" which most probably would not return relevant information to the user.

3.1.1.4 Token Normalization

After the document is split into "correct tokens" the next step would be to normalize them. *Token normalization* is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens [16]. Token normalization has as a goal to link together tokens with similar meaning. For example the token Computer should be linked to other tokens that can be synonyms of the word computer like PC, Laptop, MAC, Servers etc. So when a user searches for Computers the results of the documents that contain the token PC, Laptop, Mac and Servers should also be displayed to the user. Also some extra work should be done to remove connection between tokens that have two or more meanings. For example if you search Google for the term "Thinnest Pad in the Market" you end up with results like "Apple iPad", "Always Ultra" , "Android Tablets" and "Carefree". Now the user could have been looking for both types of pads. In order to get the proper results another token should be linked to the token Pad, for example if the user was looking for Pad/tablets he should write the query "Thinnest Android/Intel/AMD Pad". Since there is no link between android, Intel or AMD to other types of Pads the user would now get the desired results.

3.1.1.5 Stemming and Lemmatization

Most of the times the user will query for a document by typing words that are not written exactly the same in the documents he is searching for. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form [16]. For example:

Computer's, Computers, Computer, Compute -> Compute

And also different tenses of a verb should also be converted to one base form.

Was, am, are, is, have been -> be. So a search for "*android computers are fun*" would be translated into "*android compute be fun*". However stemming and lemmatization perform different functions.

Stemming usually refers to the process of removing suffixes and affixes from a certain word until it is left to its root stem, for example *unbelievable* stemmed would be *believe*. *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma* [16]. The main idea of both processes is to get a boost in information retrieval performance by subsiding variants of a keyword that otherwise would be required to be treated independently. Also index size is reduced because of the cuts in words that both lemmatization and stemming perform.

3.1.1.6 Stop Word Removal

In all the documents there are words that are unlikely to help in indexing. Those words are usually prepositions, articles, and pro-nouns. Typical stop words are prepositions, articles and pro-nouns, but the list of stop words may vary from language to language and may also vary according to different approaches in building the list of stop words. One key benefit of stop word removal is that query processing would be faster and index size of documents would be smaller. On the other side, if stop word removal does no harm to key word query searching, it really decreases retrieval relevance in phrase queries. This chapter will only focus in the stemming part of IR so the other stages like indexing and searching, ranking and evaluation will not be treated here.

3.2 Albanian Language Processing and Stemming

As discussed previously, stemming is the procedure of reducing the words to their roots. Stemming is a very complicated procedure and its difficulty varies according to the morphology of the language being used. If an English stemmer or a stemmer for languages that use Latin letters may be relatively complex, a stemmer for Chinese where the algorithm has to reduce to its roots characters like: 日本 東京 大阪 北海道 would have more difficulty. So the question that arises is why stemming is so important in information retrieval that we should go under such complicated procedures? Many theories and experiments have been performed to evaluate the efficiency and the stability of the stemming process in information retrieval. Lennon (1981) did an evaluation research about stemming techniques and how these affect the search precision, demonstrating that stemming raises the effectiveness of information retrieval. This was enough to motivate more and more researchers on stemming improvement [20].

3.2.1 Stemming Algorithms

The first ever published stemmer was written by Lovins [15].This paper was remarkable for its early date and had great influence on later work in this area. A later stemmer was written by Martin Porter [22]. This stemmer was very widely used and became the de-facto standard algorithm used for English stemming.

3.2.1.1 Lovins Algorithm

The Lovins stemming algorithm has its bases in removing the endings of a word based on a iteration-longest-match principle. Iteration is usually based on the fact that suffixes are attached to stems in a "certain order” [15]. The *longest-match* principle states that within any given class of endings, if more than one ending provides a match, the one which is longest should be removed. This principle is implemented by scanning the endings in any class in order of decreasing length. For example, if *-ion* is removed when there is also a match on *-ation,* provision would have to be made to remove *-at,* that is, for another order-class. To avoid this extra order-class, *-ation* should precede *-ion* on the list (Lovins, 1968).

Lovins defined a list of 260 endings, 21 conditions and 35 transformation rules for her algorithm. The working phases of the algorithm are like follows: First a word is taken and then an ending (out of 260) that satisfies one (out of 21) conditions is found and removed. The next step is to transform the word in case it is in irregular plural form or if it has double consonants. Also Lovins algorithm has a core rule that a stem should be 3 or more letters to be considered as such. The classical example that better demonstrates Lovins’ stemming algorithm is the procedure for stemming the word nationally. According to Lovins’ rules, this word has two endings *ionally* (leaving as stem *nat*) and *ationally* (leaving as stem n). But, because to be considered a stem from the Lovins’ algorithm the word has to be at least 3 letters, only *ionally* is accepted as an ending therefore generating the stem nat.

3.2.1.2 Porters Algorithm

Porter algorithm defines five successively applied steps of word transformation. Each step consists of set of rules in the form <condition> <suffix> -> <new suffix>. For example, a rule (m>0) EED ->EE means “if the word has at least one vowel and consonant plus EED ending, change the ending to EE”. So “agreed” becomes “agree” while “feed” remains unchanged [25].

Porter has two main differences from Lovins’ algorithm. The first one is that the rules used for suffix removal are easier and less complex. The second difference is that Porter uses a single, cohesive approach to handle the context of words differently from Lovins that used context-sensitive rules to handle the length of stems after the suffixes are removed. The algorithm is much simpler in concept with around 60 suffixes, just two recording rules and only one rule to determine if a suffix is to be removed or not. Porter avoids the rule of checking the number of the remained characters after the removal of affixes by making a consonant-vowel-consonant(called measure *m*) check after the affix has been removed [22].

3.2.1.3 Hidden Markov Models

Another statistical approach to stemmers design was used by [17] to build a stemming algorithm based on Hidden Markov Models (HMM). It doesn't need a prior linguistic knowledge or a manually created training set. Instead it uses unsupervised training which can be performed at indexing time [25]. The key idea is that an HMM is a finite model that describes a probability distribution over an infinite number of possible sequences [8]. A very clear description of HMM theory has been written in [23]. One speaks of an HMM generating a sequence. The HMM is composed of a number of states, which might correspond to positions in a 3D structure or columns of a multiple alignment. Each state 'emits' symbols (residues) according to symbol-emission probabilities, and the states are interconnected by state-transition probabilities. Starting from some initial state, a sequence of states is generated by moving from state to state according to the state-transition probabilities until an end state is reached. Each state then emits symbols according to that state's emission probability distribution, creating an observable sequence of symbols [8].

Since probability of each path can be computed, it is possible to find the most probable path (with Viterbi decoding) in the automata graph. Each character comprising a word is considered as a state. The authors divided all possible states into two groups (roots and suffixes) and two categories: initial (which can be roots only) and final (roots or suffixes). Transitions between states define word building process. For any given word, the most probable path from initial to final states will produce the split point (a transition from roots to suffixes). Then the sequence of characters before this point can be considered as a stem [25].

3.2.2 Albanian Stemmer

Most of the research for stemming algorithms has been dedicated to the English language and is based on the morphology and grammar rules of English. This because English is the most popular language and most of the researches and papers are written for this language. Applying Lovins, Porter, or Hidden Markov Model in Albanian is not a straightforward task. The morphology and grammar rules of Albanian are slightly different and thus modifications are needed for these algorithms to be applied to Albanian. There has not been much research nor many attempts to develop a stemming algorithm for Albanian. The first stemming algorithm ever was developed only in 2011 [24]. Albanian Language is very rich in inflectional paradigms making the research and development very challenging. The algorithm developed in [24] is composed of 5 steps and it is basically a dictionary based algorithm. The 5 five steps of the algorithm are described shortly below. A more detailed description can be found in [24].

Step 1: Prefix Reduction
Rules used in this step derive from the morphological analysis done for Albanian language. For example the word “pastrehë” (homeless) will become “strehë” (home).

Step 2: Suffixes that end with Vowel Reduction
Among all suffixes, there are some suffixes that end with a vowel like *–je* (*mbathje, veshje* ect.) or *–shmëri* (*ngjashmëri, gadishmëri* etc.) that are processed by this step. This step is executed before vowel removal step because if the last vowel is first, the whole suffix is not recognized by the suffix removal step.

Step 3: Vowel Reduction
There are a lot of words that are not affected by the first rule or are affected by it and give a form of the word that can be further improved by removing the ending vowel (-a, -e, -ë, -i, -o, -u), for example mbivlerë is modified by the first rule as vlerë but it still can be modified by removing the vowel. Words like dera, hëna, fletore, punëtore, mësuese, djalë, vezë, libri, teli, djalo, biro etc. that are not modified by the first rule can also be modified by this step.

Step 4: Suffix Reduction
As mentioned in the first step, suffix formation is especially the most important way of forming parts of speech in Albanian. A full list of these suffixes is given in Appendix A. Suffix removal step is divided into two steps because there are some suffixes that end with vowel and they all are considered in step 2.

Step 5: Vowel Reduction
After performing step 4, there are also some words that contain a vowel in the end, and it is better to remove those vowels. As an example, consider the word gatishmëri that from step 3 is converted as gati. This word can be processed further by removing the vowel i giving the stem gat.

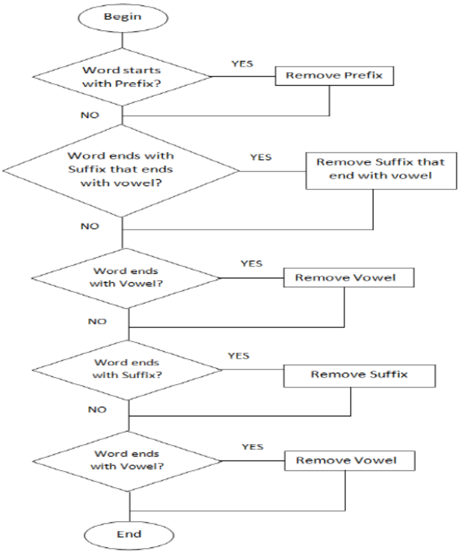


Fig. 1 Steps of the Albanian Stemming Algorithm (taken from [24])

4 Apache Hadoop

4.1 General Framework

A growing number of large companies like Google and Microsoft have implemented hundreds of special-purpose datacenters that process large amounts of raw data, such as crawled documents, web request logs, etc., to compute various kinds of derived data, such as inverted indices, various representations of the graph structure of web documents, summary of the number of pages crawled per host, the set of most frequent queries in a given day, etc. Most such computations are straightforward. However the input data is usually large and the computations have to be distributed across hundreds of thousands of machines in order to complete tasks in a reasonable amount of time [5].

To give a brief understanding on how large the data could be and why big computational power is needed let us give a simple example. Let us assume that a research will be done for analyzing the entire web. As reported on CNN [4] in October 2006 there were 100 million websites. And that number had grown by 50 million in four years (there were 50 million websites in 2004) which means that probably today there should be around 150 million websites. Let us presume that the average website has at least 10 pages and each page an average size of 50 KB then we would have: 150000000 x 10 x 200 Kb = 300 TB of data which means that we need like 300 hard disks just to save the data. The computers on average read 35-40 Mb/sec of information from disks which means that one single computer needs: 300000 (Giga bytes of information) / (35 x 60 x 60 x 24) the amount of information that can be read per day by a normal computer = 99 days which is 3^+ months just to read the web information. When the time to read the information is 3 months the time to process it could be 1 year or more.

4.1.1 Apache Hadoop

On the industry front companies such as Google and its competitors have constructed large scale data centers to provide stable web search services with fast response and high availability. On the academia front, many scientific research projects are being conducted on large datasets and powerful processing capacity delivered by supercomputers, else known as e-Science [13].This huge demand for high computational power motivates the need for cloud computing. However even in large data centers the task of processing of this large amount of data is not an easy task because there are issues such as work load distribution, failure management, cluster configuration as well as a proper programming/processing model to be taken care of. With the increasing popularity of datacenters, it is a challenge to provide a proper programming model which is capable to support appropriate access to the large scale data for carrying out computations while hiding all low-level details of physical infrastructures and among the candidates, Apache Hadoop project is one of the most popular programming models designed for this purpose [13].

Apache official presentation describes the Apache Hadoop a software library as a framework that allows for distributed processing of large data sets across clusters of computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures [1].

Apache Hadoop is composed of three main subprojects, Hadoop Common, Hadoop Distributed File System (HDFS™) and Hadoop MapReduce. This paper focuses only on HDFS, the storage, and MapReduce subprojects. Hadoop and its open source MapReduce programming have been adopted by many large companies. In 2008 Yahoo announced that it is running the largest Hadoop cluster with more than 10000 Linux nodes and they saw a 33% improvement comparing to the infrastructure they used before Hadoop [14]. Two years later in 2010 Facebook declared that they have the largest Hadoop cluster with 20PB of storage. In March 2011 Facebook declared that the size of the data storage had grown from 20PB to 30PB within one year and there was no more room to add a physical node in their current datacenter so they would have to construct a new more powerful Hadoop cluster to process the information they generate [26].

4.1.2 Hadoop Distributed File System (HDFS)

Just like with any other file system even with HDFS programmers can perform basic tasks like make directories (using *hadoop dfs –mkdir directory name* command), copy files form local file system to HDFS (using *hadoop dfs copyFromLocal /root/home/filename.extention destination folder* command) and other usual task.

HDFS is composed of master/slave architecture. A HDFS cluster should have at least a single NameNode, one master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which are used to manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. The NameNode records every change that occurs to file system by using a transaction log called the EditLog [3]. Usually nodes in cluster are distributed among several racks because of rack's capacity limitation. Nodes in the cluster communicate with each other by using TCP\IP protocol. A client establishes a connection to a configurable TCP port on the NameNode machine. The DataNodes communicate with the NameNode using the DataNode Protocol. A Remote Procedure Call (RPC) abstraction wraps both the Client Protocol and the DataNode Protocol. By design, the NameNode never initiates any RPCs. Instead, it only responds to RPC requests issued by DataNodes or clients [3].

Just like in every file system even in HDFS it is possible that data acquired from one DataNode get corrupted. There are several reasons why these data can be corrupted, like faults in network, faults in storage or even buggy software. Each client node when it receives a HDFS file runs checksums to verify the content of the file is correct. If the checksum is not correct than the client node handles this event just like a DataNode crush and retrieves the data form another DataNode that has the replica of the files [3].

4.1.3 Hadoop MapReduce

MapReduce is a programming model for processing and generating large data sets where users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs and a reduce function that merges all intermediate values associated with the same intermediate key [12]. So the main idea behind MapReduce is to treat all the data you want to analyze as <key,value> pairs. For example in social networking we can have <user_id, list of friends> key value pairs whereas in shortest path algorithms we can have <cities, list_of_other_cities_connected>. The MapReduce programming model was invented by Google and as the authors in [12] admit it was used in more than 10000 applications at Google since a system based in MapReduce was built in 2003. The programs include text processing, machine learning, and statistical machine translation. Apache foundation, in their Apache Hadoop project, has developed and maintains an open source version of the MapReduce programming. MapReduce programming model works as follows:

"An input of <key/value> pairs is given to a Map function and it generates a list of <key/value> pairs as output. The type of output key and value key do not have to be the same as the keys of the input:

$$map :: (key_1, value_1) \Rightarrow list(key_2, value_2)$$

A Reduce takes a <key/list_of_values> input and generates a list of new values as output:

$$reduce :: (key_2, list(value_2)) \Rightarrow list(value_3)$$

A MapReduce application has two phases that are executed in parallel. In the first phase, all map operations can be executed independently from each other. In the second phase, each reduce operation may depend on the outputs generated by any number of map operations and just like them reduce operations can also be executed independently [13].

To better understand how map reduce works let us consider the example presented in [12] for counting the occurrence of each word in a large collection of documents. In the pseudo code below

```
map(String key, String value):
// key: document name
// value: document contents
for each word w in value:
EmitIntermediate(w, "1");
reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int result = 0;
for each v in values:
result += ParseInt(v);
Emit(AsString(result));
```

The map function will create a list of words together with the number of occurrences of each word. On the other hand the reduce function will sum together all the counts created for each word and will produce the result.

5 Analysis, Design and Implementation of the Plugin

In this section we describe the development of the plugin and the deployment on the cluster.

5.1 Analysis

In order to convert the stemming algorithm of [24] (referred to as JStem from now onwards) from an algorithm that runs in a single machine to a MapReduce algorithm that runs in a cluster, the first decision to take is what will be the <key, value> pairs of the Map Phase. These pairs are fundamental since these will later be used by the Reduce phase to generate <key, list of values> pairs as output.

The JStem algorithm works as follows: it takes a text file as input and then uses a buffered reader to read the file. It uses a String tokenizer to break the lines of the input file into tokens and for each token the algorithm calls the stem procedure (that takes a string as a parameter). The stem procedure first analyzes the input and then returns its root (the stemmed word) thus creating a <word, stemmed_word> pair for each word in the file that was given as input. These pairs (keys) are then inserted into a TreeMap structure where they are sorted alpha betically. After the process of reading and sorting is over then the algorithm iterates through each key of the TreeMap and writes into a file each <word, stemmed_word> pair.

After analyzing the source code of the JStem algorithm it was observed that a MapReduce paradigm could be implemented upon the original JStem algorithm without changing the way it finds the root of the words but instead changing the procedure for calling the stemming procedure and then collecting back the results. Given that the original design of the JStem algorithm was based on generating a <word, stemmed_word> set of values, the integration of the algorithm with the MapReduce programming model would be possible by respecting the requirements of the given platform.

5.2 Design and Implementation

In order to convert the JStem algorithm into MapReduce programming model 3 classes were added to the JStem class which is the class that holds the logic of stemming. The classes that were added are the Map class that holds the logic of the Map phase, the Reduce class that holds the logic for the Reduce phase and the HadoopStemmer class that holds the logic for execution and job configuration in apache Hadoop.

The Map class extends the MapReduceBase and implements the MapReduce class. Individual Map tasks take as an input <key,value> pairs and then transform these pairs into intermediate <key,value> pairs. The intermediate pairs can or cannot be of the same type as the input pairs. All the intermediate pairs related to one key are grouped by the framework and are then submitted to the Reducer that determines what the output will be. In the mapper phase intermediate pairs are created and are grouped by the root of the word, for example cooking and cooker have the same root cook and two intermediate pairs <cook,cooking> and <cook,cooker> are created.

The Reduce class extends the MapReduceBase class and implements Reducer. Following the same logic as the Map class it has four generic types of type Text. After the Map phases are finished the framework calls the reduce function. The reducer shuffles the intermediate values and then orders them alphabetically and thus producing the final output.

In the HadoopStemming class we define the properties for the job that define different parameters and tell the HadoopFramework which is the Map class, which is the Reduce class, what is the format of the input and output, etc.

5.3 Running the Plugin in the Hadoop Cluster

In order to run the plugin in the Hadoop Cluster first some configuration files should be changed according to the developer's needs. The configurations are stored at /usr/local/Hadoop/conf/ folder. The file core-site.xml determines the port that will be used between nodes to communicate with each other and tell the other nodes who the authority master node is. It also defines the path of Hadoops' temporary file storage.

The file mapred-site.xml defines the maximum number of Map tasks that are run in parallel. In this case the maximum is set to 800. This number is then divided by the number of nodes in the cluster do define the number of Map tasks that will be run on each node. The same logic is used for the number of the Reduce phases. Since the Reduce phase is a little more complicated it requires more resources from the computers and as such it is advisable to keep the number of reduce task to 4 times the number of nodes in the cluster. If set to 0 the Reduce phase will not run and the outputs will be unsorted.

Another important configuration file is hdfs-site.xml. In this configuration file the user defines the default block replication. Which means the user determines how many copies of its data he wants to distribute in the cluster. The value should be at maximum the same as the number of nodes in the cluster.

6 Evaluation

In this section we present the experimental evaluation of the developed plugin.

6.1 Experimental Setting and Cluster Creation

In order to run and test the MapReduce plugin first a cluster was created. To create the cluster 10 computers (Dell Optiplex 360 with Intel dual core 2.2 GHz processor and 4 Gb of RAM) running Ubuntu Linux 11.10 were used. Also Java version 1.6 was used as framework. For creating the cluster, the [19] tutorial was used as a guideline although some changes were made to adapt the cluster to the experiment's needs. There were 10 computers in the cluster where one of the computers acted as the master (with host name master) and the nine other computers were slaves (with hostnames slave1, slave2 … slave9). The computers were connected to each other using a 24 port switch and using static IP configuration. IP V6 was disabled on all machines as required by Hadoop in order to run smoothly.

6.2 Dataset

In order to best evaluate the performance of the algorithm when running on a single machine compared with the performance of the same algorithm running in a cluster, a data set consisting of large files was created. To create the data set several Albanian books (books of history, science, research and poetry) were downloaded and then converted to plain text. Because the original algorithm created in [24] accepts only one text file as input a total of 6 files with sizes 6.9 MB, 13.8MB, 20.7MB, 27.6MB, 34.5MB, 41.4MB were created to be used as input , which is basically increasing the size of the original file by 6.9 MB.

6.3 Evaluation Setting and Experimental Results

To better understand the power of MapReduce distributed programming a slight modification was made to the original algorithm to sort the stemmed words alphabetically after stemming them. The files created to be used as input would be executed 3 times each in a single computer, 3 times each in the cluster running 2 nodes and 3 times each in the cluster running 5 nodes. The average time of each run was recorded and then stored in a table to be later used for evaluation and time performance.

The first test was performed for the JStem stemmer developed in [24], with a text file with size 6.9 MB. In three runs it was observed that the average execution time in a single PC was 15.2 seconds. The same test was then performed on the Hadoop Stemmer which processed the file on an average time of 27.9 seconds in a 2 node cluster and 27 seconds on a 5 node cluster.

The JStem continued to run faster even for a 13.8 MB file where in 3 runs it recorded an average time of 49 seconds compared to the 58.9 seconds of the 2 Node Cluster and 56.1 seconds of the 5 Node Cluster. However things started to change when the file size was increased from 13.8 MB to 20.7MB where the JStem recorded an average time of 159.1 seconds while the 2 and 5 Node Hadoop cluster recorded an average time of 97 and 94 seconds respectively. Then the file size was increased with another 6.9MB taking it to 27.6MB in total. The Hadoop

Cluster continued to perform better than the JStem where it recorded and average time of 154 seconds for 2 Node Cluster and 150.1 seconds for 5 Node Cluster whereas the JStem recorded an average time of 510.1 seconds.

When the file size was increased by another 6.9 MB (34.5 MB in total) the sorting process of the JStem algorithm ran out of memory (basically ran out of RAM) and threw an out of memory exception while the Hadoop Cluster completed the job with no problems.

The Hadoop Cluster faced some problems when the file size was increased to 41.4 MB where one of the nodes failed during the Reduce Phase of the algorithm. However the process was not interrupted because the other nodes that had the same replica of the file took over and the process completed successfully.

6.4 Analysis and Interpretation of Results

The results gathered from the tests are presented in Table 1 and then some analyses are performed with the data to see the behavior and time complexities of both the JStem and the Hadoop Cluster.

Table 1 Running time of JStem compared with MapReduce in 2 Nodes and 5 Nodes Cluster

File Size (MB)	JStem (seconds)				Hadoop with 2 Nodes (seconds)				Hadoop with 5 nodes (seconds)			
	Run1	**Run2**	**Run3**	**Average**	**Run1**	**Run2**	**Run3**	**Average**	**Run1**	**Run2**	**Run3**	**Average**
6.9	15.17	15.2	15.2	15.2	27.8	27.3	28.6	27.9	27.1	26.5	27.3	27.0
13.8	48.6	48.4	50	49.0	58	60.1	58.5	58.9	56.4	56.2	55.8	56.1
20.7	158.6	159	160	159.1	96.7	97.5	96.9	97.0	94.3	94.2	93.4	94.0
27.6	508.7	510.4	511	510.1	153.5	154	154	154.0	150	151	150	150.1
34.5	crash	crash	crash	crash	248.8	251	250	250.1	239	239	236	237.9
41.4	crash	crash	crash	crash	464.8	466	464	465.1	449	445	447	447.0

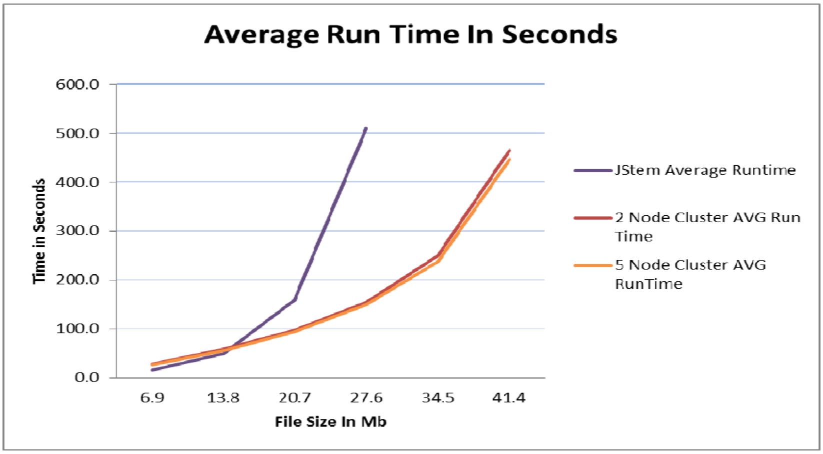


Fig. 2 Average runtimes of all the algorithms

The runing time complexity of the JStem algorithm with respect to the filesize is presented in Figure 3. Whenever the file zise is increased by 6.9MB the execution time of the JStem is nearly trippled. To prove that some trend lines were drown to analyze the bahavior of the JStem algorithm. The results can also clearly be seen at execution time table. As we can observe in the chart below the time complexity is aproximately y = 18.391x^3 - 56.707x^2 + 59.707x.

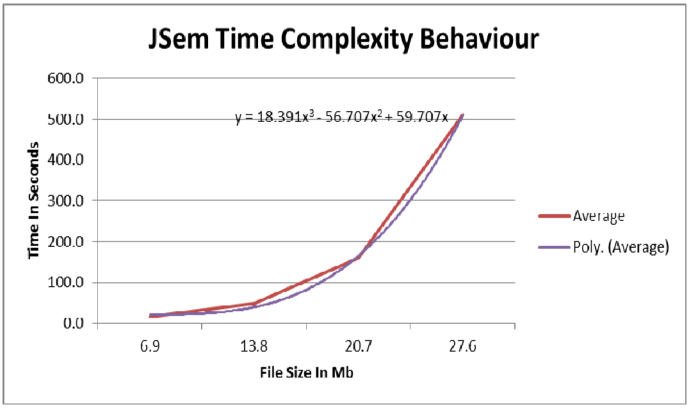


Fig. 3 JStem time complexity

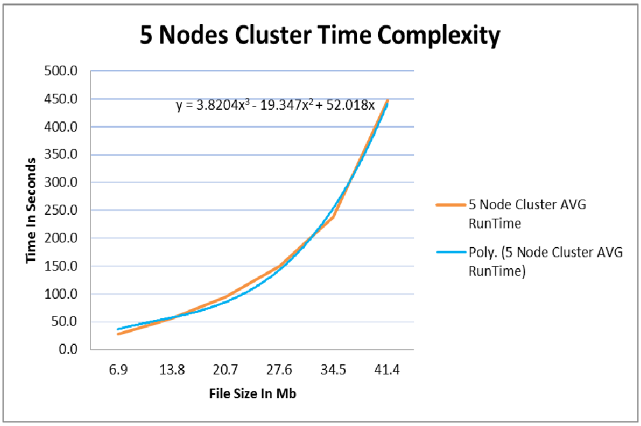


Fig. 4 MapReduce 5 Nodes cluster time complexity

When comparing the first 2 run times of the JStem and the algorithm running on the stemmer we can observe that the JStem runs faster. However for each increase in 6.9 MB of the file size the execution time of the algorithm running in the cluster is doubled whereas that of the JStem is tripled. That is the reason that when processing files larger than 20.7MB the algorithm in the cluster is faster.

7 Conclusion and Future Work

In this chapter we have presented an Apache Mahout plugin for a stemming algorithm making possible to execute the algorithm in a cloud computing environment. We have investigated the performance of the algorithm in the cloud and shown that the new approach based on MapReduce significantly reduces the execution time of the original algorithm over a large dataset of text documents. In addition, sorting the results alphabetically in very acceptable time was another benefit provided by MapReduce programming model, which was observed during the execution of the algorithm in the Hadoop Cluster.

The presented work can be improved in several directions. First, the cluster was constructed using normal computers and not dedicated server machines which if used could reveal the full potential of the Hadoop framework. During the testing phase of the plugin a lot of network failures occurred which limited the power of the cluster to 5 out of 10 configured nodes. This means that better network equipment and optimization could be implemented in future work.

Interesting future work regards the exploitation of the already built cluster and integrating other language processing tools such as part-of-speech tagging, parsing, named entity recognition etc. Moreover, we are currently working on integrating a full text classification and categorization engine for Albanian in a cloud computing framework.

References

1. Apache Hadoop Foundation, Apache Hadoop (2011), from Hadoop `http://hadoop.apache.org/` (retrieved November 15, 2011)
2. Bakshi, K.: Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions. Cisco Systems, Inc., California (2009)
3. Borthakur, D. Hadoop Distributed File System, from Hadoop Distributed File System (2012), `http://hadoop.apache.org/` (retrieved March 2012)
4. CNN. CNNTECH (2006), from CNN `http://articles.cnn.com/2006-11-01/tech/100millionwebsites_1_web-site-cern-tim-berners-lee?_s=PM:TECH` (retrieved November 23, 2011)
5. Dean, J., Ghemawat, S.: Map Reduce: Simplified Data Processing on Large Clusters. Google, Inc. (2004)
6. Dialogic, I. (2011) from Dialogic Inc. `http://www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf` (retrieved November 22, 2011)
7. Dikaiakos, M.D., Pallis, G., Katsaros, D., Mehra, P., Vakali, A.: Cloud Computing, Distributed Internet Computing for IT. Internet Computing 13(5) (2009)
8. Eddy, S.R.: Hidden Markov Models. Current Opinion in Structural Biology 6, 361–365 (1996)
9. Habert, B., Adda, G., Adda-Decker, M., Marueuil, P.B., Ferrari, S., Ferret, O., et al.: Towards Tokenization Evaluation. In: Proceedings of LREC (1998)

10. Han, L., Saengngam, T., van Hemert, J.: Parallel Data Intensive Applications in Cloud: A data mining use case study in the Life Sciences (Extended Abstract). In: UK-eScience AHM Meeting, in Cardiff (2010)
11. Jackson, P., Moulinier, I.: Natural language processing for online applications: Text retrieval, Extraction and Categorization. Language 5(1), 178–178 (2002)
12. Jeffry, D., Sanjay, G.: MapReduce: A Flexible Data Processing Tool. Communications of the ACM 53(1), 72–77 (2010)
13. Jin, C., Buyya, R.: MapReduce Programming Model for.NET-Based Cloud Computing. In: Sips, H., Epema, D., Lin, H.-X. (eds.) Euro-Par 2009. LNCS, vol. 5704, pp. 417–428. Springer, Heidelberg (2009)
14. Kunz, C.: Yahoo Webmap Research (2008), from Yahoo `http://www.research.yahoo.com/files/YahooWebmap.pdf` (retrieved January 25, 2012)
15. Lovins, J.B.: Development of a Stemming Algorithm. Mechanical Translation and Computational Linguistics 11, 22–31 (1968)
16. Manning, C.D., Raghava, P., Schütz, H.: An Introduction To Information Retrieval. Cambridge University Press, Cambridge (2009)
17. Melucci, M., Orio, N.: A novel method for stemmer generation based on hidden Markov. In: Proceedings of CIKM 2003, 12th International Conference on Information and Knowledge Management. ACM, New York (2003)
18. Mell, P., Grance, T.: NIST Working Definition of Cloud Computing. National Institute of Standards and Technology, Information Technology Laboratory, Rockville, Columbia (2009)
19. Noll, G.M.: Running Hadoop On Ubuntu Linux (2011), from Michael Noll Hadoop Tutorials `http://www.michael-noll.com/tutorials/` (retrieved January 10, 2012)
20. Ntais, G.: Development of a Stemmer for the Greek Language. MSc Thesis, Stockholm University (2006)
21. Peng, B., Cui, B., Li, X.: Implementation Issues of A Cloud Computing Platform. IEEE Data Eng. Bull. 32(1), 59–66 (2009)
22. Porter, M.: Porter Stemmer, `http://tartarus.org/~martin/` (retrieved November 15, 2011)
23. Rabiner, L.: A tutorial in Hidden Markov Models and Selected Applications in Speech Recognition. Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco (1990)
24. Sadiku, J.: A Novel Stemming Algorithm for Albanian in a Data Mining Approach for Document Classification. MSc thesis, University of New York, in Tirana (2011)
25. Smirnov, I.: Overview of Stemming Algorithms. Mechanical Translation. DePaul University, Chicago (2008)
26. Yang, P.: Facebook Blogspot (2011), from Facebook Blogspot (retrieved January 5, 2012)
27. Zhang, S., Zhang, C., Yang, Q.: Data Preparation for Data Mining. Applied Artificial Intelligence 17(5-6), 375–381 (2003)

A Recommendation System for Browsing of Multimedia Collections in the Internet of Things

F. Amato, A. Mazzeo, V. Moscato, and A. Picariello

Abstract. Exploring new applications and services for mobile environments has generated considerable excitement among both industries and academics. In this paper we propose a context-aware recommender system that accommodates user's needs with location-dependent multimedia information available in a mobile environment related to an indoor scenario. Specifically, we propose a recommender system for the planning of browsing activities that are based on objects features, users' behaviours and on the current context the state of which is captured by apposite sensor networks. We present the features of such a system and we discuss the proposed approach.

1 Introduction

Several years ago, the Economist reported that people read about 10MB worth of material a day, hear 400 MB worth of audio data a day and see about 1 MB of information every second. This is the main reason for which web gurus claim that the Web is leaving the era of *search* and entering the one of *discovery*, where "search is what you do when you're looking for something", while "discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you" [1,2]. In other words, the new cool research topic in information technology seems to be "recommendation" instead of the more classic "search".

Generally speaking, a recommender system is a tool that is able to make personalized suggestions to users by analyzing previous interactions among users and the system itself. Recommender systems use the opinions of a community of users and items features to help individuals to more effectively identify content of interest from a gigantic set of choices.

In the last few years, recommender systems have gained significant attention in the Internet research community, due to the increasing availability of huge collections of data items (e.g., digital libraries, virtual museums, news archives, shopping catalogues and so on), and the pressing need for applications to provide users with targeted suggestions to help them navigate this ocean of information.

F. Amato · A. Mazzeo · V. Moscato · A. Picariello
Dipartimento di Informatica e Sistemistica,
University of Naples "Federico II"
e-mail: `{flora.amato,mazzeo,francesco.moscato,picusg}@unina.it`

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 391–411.
DOI: 10.1007/978-3-642-34952-2_16

However, despite their increasing application in the virtual world, not enough effort has yet been devoted to the adoption of recommenders in the real world, i.e. in the context of the so-called *Internet of Things*.

The Internet of Things is conceived as a new form of distributed computing, in which the communication takes place between humans and a network of physical objects [3]. The classical view of ubiquitous computing, which envision anytime, anyplace connectivity for anyone, is now enriched with the concept of connectivity for anything, hence enabling new forms of communication between people and things. The recent technical advances in hardware and in software research and development are contributing to the rapid diffusion of the Internet of Things paradigm.

Device miniaturization and the availability of wireless communication technologies enable short-range mobile radio transceivers to be embedded into a wide set of smart devices, able to sense parameters of interest from the real world and represent them in the virtual world. Examples are small devices equipped with RFID tags or smart sensors constituting the so-called *Wireless Sensor Networks*. At the same time, several middleware platforms are emerging, enabling the development of software systems able to manage the smart interconnection of people and things.

Despite these efforts, the development of advanced services for the Internet of Things is still complicated by the high dynamicity of the system, which has to deliver meaningful information to users in real-time, depending on their movements in the physical space, on their interaction with smart sensors, and accounting for their past choices and current needs. In particular, in this kind of systems, useful information for users is dependent on several parameters, such as: user location, user interests, network performance, semantics of the contents, multimedia information features, other context information, and so on.

In addition, it should be useful for a user to receive a set of suggestions for navigating the physical space in terms of *browsing activities*. To this aim, we see the opportunity of using recommender systems as a key element to orchestrate user movements in the physical space, while taking the burden of acquiring and processing the needed contextual information from surrounding smart sensors.

As motivating example, we can consider the case of a real museum (i.e. Uffizi Gallery in Florence) that offers by a WIFI connection a web-based access to a multimedia collection containing: digital reproductions of paintings, educational videos, audio guides, textual and hypermedia documents with description of authors and paintings. In order to make the user's experience in the museum more interesting and stimulating, the access to information should be customized based on the specific profile of a visitor, which includes learning needs, level of expertise and personal preferences, on user effective location in the museum, on the "paintings similarity" and on information about the context in terms of number of persons for each room, room fitness, network performances, etc.

Let us consider a user visiting the room 18 of the museum containing some paintings depicting by Alberto Vasari and suppose that he is attracted, for example, by the painting entitled "Allegory of the Immaculate Conception" (Figure 1a). It would be helpful if the system could learn the preferences of the user (e.g. interests in paintings depicting the "Holy Mary" subject), based on the user current behaviour and past interactions of other visitors, and predict his future needs, by suggesting other paintings (or any other multimedia objects) representing the same or related subjects, depicted by the same or other related authors, or items that have been requested by users with similar preferences.

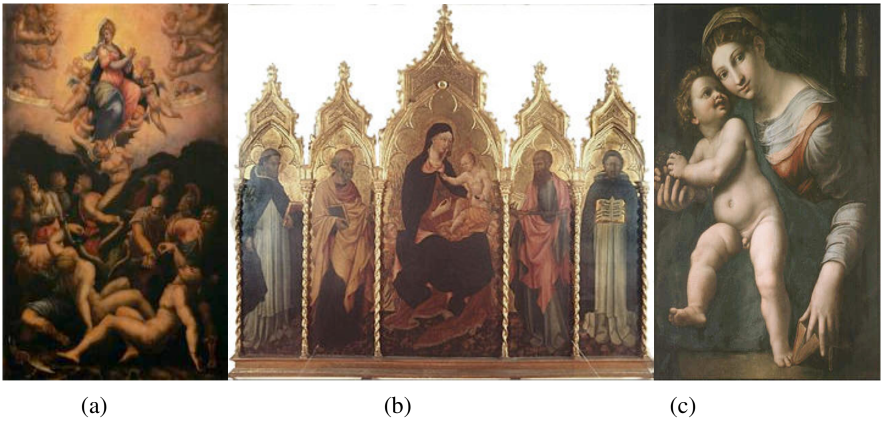
As an example, the user who is currently observing the Vasari's painting in Figure 1.a might be recommended to see in the same room a Giulio Lippi painting entitled "Madonna col Bambino" (Figure 1.c), that is quite similar to the current picture in terms of colour and texture, and in the room 5-6 "Madonna col Bambino e Santi" by Giovanni di Paolo (Figure 1.b), that is not very similar in terms of low level features but is more similar in terms of semantic content.

Moreover, if in the past a lot of visitors after having seen the Vasari's painting visited the room containing the special "Pontormo and Rosso Fiorentino" collection and such a room is not too crowded, the system could suggest visiting it and recommending some paintings inside. If it is requested, the system could suggest by using museum maps the way to reach each room from the current position. During the visit, a user by his mobile device could read multimedia documents related to authors or paintings that he/she is watching and listen audio guides available for different languages.

In such a context, we specifically tried to provide an answer to the following main research question: how to perform a planning of browsing activities in a real indoor scenario that is based on objects features, users' behaviours and on the current context the state of which is captured by apposite sensor networks?

In this paper we aim at proposing a novel approach to recommendation of browsing activities for multimedia collections in the Internet of Things. The approach is based on: (i) an importance ranking method that strongly resembles the well-known PageRank ranking system, (ii) an ontological representation of the observed context which state is captured by means of sensor networks deployed in the real environment.

We model recommendation as a social choice problem, and propose a method that computes customized recommendations by combing, in a novel and original way, intrinsic features of the objects, past behaviour of individual users, behaviour of the users' community as a whole and information on the context.



(a) (b) (c)

Fig. 1 Paintings depicting Holy Mary

The context is gathered from a set of smart sensors deployed in the environment. In particular, in this chapter we focus on the user location and on the information about the co-presence of users, in order to estimate if a given environment is crowded.

The chapter will be organized as follows. Section 2 discusses the state of the art of recommender systems and of the context-aware and multimedia related applications. Section 3 illustrates a functional overview of the proposed recommender system. Section 4 presents the proposed activity planning strategy describing both the used model of context and the recommendation technique. Section 5 describes some implementation details, while section 6 reports preliminary experimental results; finally, section 7 gives some concluding remarks and discusses future work.

2 Related Work

In the most common formulation, the *recommendation problem* is the problem of estimating *ratings* - sometimes called also *utilities* - for the set of items that has not yet been seen by a given user [4].

In *Content Based recommender systems* [5], the utility r^i_j of item o_j is estimated using the utilities $r(u_i, o_k)$ assigned by the user u_i to items o_k that are in some way "similar" to item o_j. For example, in a movie recommendation application, in order to recommend movies to user *u*, the content-based recommender system tries to understand the commonalities among the movies user *u* has rated highly in the past (specific actors, directors, genres, subject matter, etc).

Then, only the movies that have a high degree of similarity to the user's preferences would be recommended.

One of the main drawbacks of these techniques is that they do not benefit from the great amount of information that could be derived by analyzing the behavior of other users. Moreover, the content must either be in a form that the related features can be automatically parsed and if two different items are represented by the same set of features, they are indistinguishable. Finally, a subtle problem is that the system can only recommend items that are similar to those already rated by the user itself (*overspecialization*).

Collaborative Filtering [6] is, in the opposite, the process of filtering or evaluating items using the opinions of other people. Thus, unlike content-based recommendation methods, collaborative systems predict the utility of items r^i_j for a particular user u_i based on the utility $r(u_h,o_k)$ of items o_k previously rated by other users u_h "similar" to u_i. It takes its root from something human beings have been doing for centuries: sharing opinions with others. These opinions can be processed in real time to determine not only what a much larger community thinks of an item, but also develop a truly personalized view of that item using the opinions most appropriate for a given user or group of users[7].

The main problem behind collaborative filtering clearly is to associate each user to a set of other users having similar profiles. In order to make any recommendations, the system has to collect data mainly using two methods: the first one is to ask for explicit ratings from a user, while it is also possible to gather data implicitly logging actions performed by users. Once the data has been gathered, there are two basic ways of filtering through it to make predictions. The most basic method is *passive filtering*, which simply uses data aggregates to make predictions (such as the average rating for an item) and each user will be given the same predictions for a particular item (e.g. *digg.com*). In the opposite, *active filtering* uses patterns in user history to make predictions obtaining user-specific and context-aware recommendations (e.g. *Amazon*).

Collaborative systems have their own limitations that can be grouped under the name of the *cold start problem* that describes situations in which a recommender is unable to make meaningful recommendations due to an initial lack of ratings thus degrading the filtering performance. We can have three scenarios: *new user*, *new item* and *new community*.

Content-based filtering and collaborative filtering may be manually combined by the end-user specifying particular features, essentially constraining recommendations to have certain content features.

More often they are automatically combined in the so called *hybrid approach* [8] that helps to avoid certain limitations of each method. Different ways to combine collaborative and content-based methods into a hybrid recommender system can be classified as follows: (i) implementing collaborative and content-based methods separately and combining their predictions; (ii) incorporating some content-based characteristics into a collaborative approach; (iii) incorporating some collaborative characteristics into a content-based approach; (iv) constructing a general unifying model that incorporates both content-based and collaborative characteristics.

More recently, the discussed strategies have been extended to multimedia applications (e.g. multimedia repositories, digital libraries, multimedia sharing system, etc…) with the aim of considering in a more effective way multimedia content of recommended objects, both in terms of low-level and high-level characteristics (i.e. multimedia features and semantics), in the recommendation process together with user social behavior and preferences.

For what content-based techniques concerns, Maidel [9] proposes a method that exploits some ontologies for ranking items' relevancy in the electronic paper domain, while in [10] a content based filtering has been applied to music data using decision trees. In the framework of multimedia sharing system, Musial et al. [11] introduce a recommender system that uses two ontologies (one for multimedia objects and one for users) in the context of a photo sharing system. To generate suggestions a new concept of "multirelational" social network was introduced, covering both direct as well as multimedia object-based relationships that reflect social and semantic links between users. Finally, Manzato [12] proposes a content-based recommender architecture which explores information that is available at the time users enhance content in order to capture a certain level of semantic information from the multimedia content and from user preferences that is at the base of their video recommender system.

Among collaborative-filtering proposals, Baloian et al. [13] propose a collaborative recommender system, which suggests multimedia learning material based on the learner's background and preferences. Kim et al.[7] propose a collaborative filtering-based multimedia contents recommender system in P2P architectures that rates multimedia objects of nearest peers with similar preference through peer-based local information only.

As hybrid solutions, the *uMender* system [14] exploits context information, musical content and relevant user ratings to perform music recommendations on mobile devices. Knijnenburg [15] proposes a user-centric approach to media recommendations that exploits subjective and objective measures of user experience, while users interact with multimedia data. Another example of hybrid approach, implemented in *MoRe*, a movie recommendation system, has been described in [16]. Finally, low and high level features have been used to define the similarity among multimedia items in [17]; this measure is then used to compare browsing patterns of past users in order to identify users with similar browsing behavior.

As we can note, the majority of approaches to recommendation in the multimedia realm exploits high level metadata - extracted in automatic or semi-automatic way from low level features - that are in different manners correlated and compared with user preferences, usually mapped in the shape of ontologies.

These approaches suffer from some drawbacks:

- it is not always possible to extract in an automatic and effective way useful high level information from multimedia features (automatic annotation algorithms have not always high performances);
- for some kinds of multimedia data there is not a precise correlation between high and low level information (e.g. in images the concept of "moon" is related to a region with a circular shape and white colour with a given *uncertainty*);
- there are not always available explicit and useful information (knowledge) about user preferences (e.g. usually a user can retrieve information from a multimedia system without the necessity of a registration, as in "Youtube" or "Flickr");
- in the recommendation process and for particular kinds of multimedia data sometimes it is useful to take into account features of the objects that user is currently observing as content information (e.g. the main colours of a painting are often an indication of the related artistic movement or school).

The importance of *context information* has been widely recognized in many disciplines: e-commerce, data mining, information retrieval and ubiquitous mobile computing, computational social science [18,8]. In the area of recommendation systems, however, the vast majority of existing approaches focuses on the recommendation of the items most relevant to users without taking into account any kind of contextual information such as time, place, weather and proximity to other people (social contact or proximity). Only in the last few years, the use of additional contextual information in the recommending process has brought to the introduction of the *Context-aware Recommender Systems* (*CARS*).

It was shown that CARS provide more relevant predictions in different scenario [19]. CARS approaches can be divided into three main categories [4]. In the *Contextual Pre-filtering* techniques [6] context information are used to initially select the set of relevant items, while a classic recommender is used to predict ratings. In the *Contextual Post-filtering* approaches [20] context are used in the last step of the recommending process to "contextualize", for each user, the output of a traditional recommender. Finally, in *Contextual Modelling* tools [21] additional information is used directly in the rating estimation process.

Co-clustering and, more in general, (high-order) matrix factorization techniques has been widely used both in classic and context-aware recommender systems [19]. Additional information coming from unstructured (i.e. folksonomies) and structured (i.e. taxonomies) knowledge bases has been leveraged to influence the co-clustering process [22, 23].

Summing up, our approach can be classified as a hybrid recommendation strategy that incorporates some content-based characteristics into a collaborative strategy. It exploits system logs to implicitly derive information about individual users and the community of users as a whole, considering their past browsing sessions as a sort of unary ratings.

Similarly to collaborative filtering, it is a sort of active filtering strategy in which past browsing sessions, mapped in the shape of a graph, determine the most suitable items (candidates) to be recommended.

Similarly to information retrieval and filtering content-based approaches, it considers as relevant content for the recommendation the features of the object that a user is currently watching and supposes the existence of a-priori knowledge about metadata values and their relationships (i.e. taxonomy is used to define high-level concepts).

The system considers in separate ways multimedia low and high level information, both contributing at determining the utility of an object in the recommendation process. Then, the context information is used as a "pre-filter" to decide the set of candidates for the recommendation and as "post-filter" to arrange recommended items depending on the context evolution.

Finally, it is possible to notice that information about users is not explicitly considered in the recommendation process. However, user characteristics can be learned during browsing sessions using rule discovery or data mining approaches [24, 25] and they thus can be exploited to improve the effectiveness of recommendations, providing personalized contents.

3 System Overview

Figure 2 shows at a glance a functional overview of our recommender system. It is composed of several components, described in the following.

Context Provider - It captures the data from the various heterogeneous informative sources, such as user mobile devices, sensor networks, RFID sensors, video camera, etc… It collects the sensed environmental data as temperature, lighting, humidity and information about people that are effectively present in a given room and data about user location.

Knowledge Base – It includes a set of repositories devoted to store static information that contribute to define the context such as data about museum's art collection, museum cartography and users profiles. In particular, repositories for the following typologies of information are provided:

- ***Art*** – a set of repositories storing information about the museum's art collection, the list of the exposed works, their characteristics, expresses both in terms of low-level features, such as shapes, textures, colours, and a high-level features, as authors, descriptions, concepts and content associated with it, typology, techniques, similarities with other (exposed or not) works.
- ***Cartography*** - a set of repositories storing cartographical information about museum rooms maps and the location of the exposed works.
- ***Users*** - a set of repositories storing information about users profiles as past user preferences and behaviours.

Information Integrator – It is responsible for the integration of heterogeneous information coming from different types of repositories. It provides an interface for querying the Knowledge Base, integrating information to be outputted to the user.

Context Synthesizer – It allows combining information coming from the Context Providers with the ones related to the Knowledge Base in order to provide aggregate information about the context state. It performs reasoning activities for combining the heterogeneous data, allowing the composition of complex assertions as: "in room 1 there are 10 persons", "the temperature in room 3 is 30°C", "users 1 and 2 are watching the *Monnalisa* picture in room 5" and "user 1 visited room 4 before room 5".

Context History - The repository contains historical information on the context state, i.e. a context *data warehouse* designed with respect to three main analysis dimensions: time, space and users.

Context Analyzer - It is a software agent for querying the context history repository.

Recommender System – It is the software module aiming at proposing to users the objects of interest by means of the recommendation approach that will be described in the next Section.

Activity Planner – It is the system core. For each user, on the base of information about the current and past context, and considering the list of recommended object, the component dynamically and automatically proposes a set of browsing activities that maximizes user satisfaction. We remark that the role of the Activity Planner is the central focus of the proposed architecture: it collects context information from the Context Analyzer, Context Synthesizer and Recommender System, combining it in order to plan the next activity to be suggested to the user, by means of the Activity Deliverer.

Activity Deliverer - This component aims at delivering browsing activities to each user in a format that will depend on the user profiles and devices.

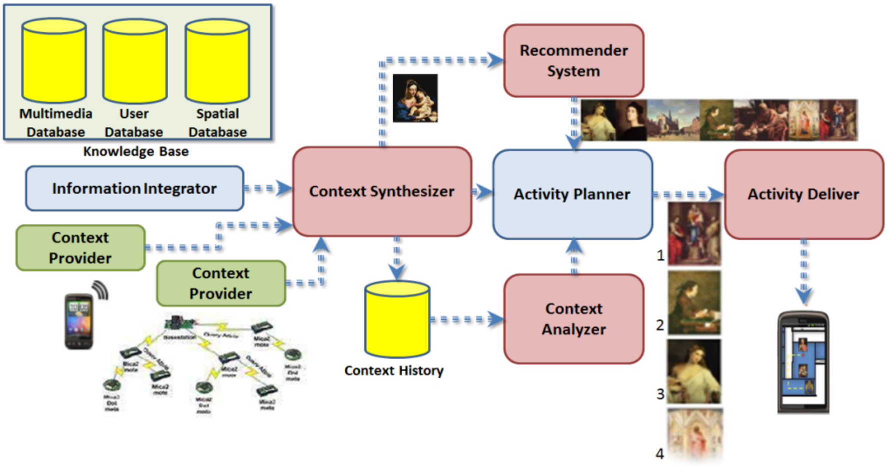


Fig. 2 The System Architecture

4 The Recommendation Model

Building an effective multimedia recommender to support intelligent planning of browsing activities in indoor scenarios implies the capability to reliably identify the “objects” and “actions” in the observed context, which are most likely to satisfy the interests of a user at any given point of his/her exploration.

At each change of the context state, the user chooses an activity, by either accepting one of the recommendations, or following personal needs. Given the following premises, we then need to address some fundamental questions:

1. How can we model the context and the related browsing activity?
2. How can we select a set of activities that can fulfil recommendation purposes?
3. How can we rank the set of candidates?

4.1 Modelling the Context

We proceed to model the reference context by using an ontological approach. In particular, in a similar manner to CONON[27], we adopt two ontologies: an *upper level ontology* able to describe general concept of the context and a *lower level ontology* in which the concepts are particularized in a given application domain: an indoor museum, in our case of study.

In the upper level ontology, aiming at describing the user environment, the context is modelled with five elements: **Entity**, **IndoorLocation**, **Device**, **Activity** and **TimeInterval**. Figure 3 reports the T-Box formalizing the user context. Following Dey ed Abowd [4] definition, the **Entity** concept is the set of persons and objects considered relevant for the applications. The relevant elements of the domain are then people and objects contained in it, corresponding to **Person** and **Object** classes.

The **IndoorLocation** concept has been designed to model the position information of indoor sites. This concept is specialized into two categories: **Room** for specifying a particular room or area to which associate a proper identifier, and **InPosition** for stating geographical location, characterized by spatial coordinates values.

The **Location** concept is associated to the **EnvCondition** concept; it identifies the set of environmental conditions of a specific location, by specifying some attributes, such as temperature, light, humidity, noise and pressure. These attributes are collected by proper sensors. They are placed in monitored area and provide environmental information about current context. The proprieties of the sensors are modeled by the **Device** Concept. This concept is specialized in two typologies of instruments, sensors for physical phenomena measurement (**PhysicalSensor**) and mobile devices for acquiring users location (**MobileDevice**).

The devices are identified by the attribute *has-id*, and associated to a particular entity, for example the owner or the object which is related to the acquired data, by means of the role *isAssociatedWith*.

$$Context \sqsubseteq Entity \sqcup IndoorLocation \sqcup Device \sqcup Activity \sqcup TimeInterval$$
$$Entity \sqsubseteq Person \sqcup Object$$
$$IndoorLocation \sqsubseteq Room \sqcup InPosition$$
$$Device \sqsubseteq PhysicalSensor \sqcup MobileDevice$$
$$Activity \sqsubseteq ScheduledActivity \sqcup DeducedActivity$$
$$Device \sqsubseteq \forall HasId.Integer$$
$$Device \sqsubseteq \exists IsDetecting.Activity$$
$$Device \sqsubseteq \forall IsLocatedIn.IndoorLocation$$
$$Device \sqsubseteq \exists IsAssociatedIn.Entity$$
$$Entity \sqsubseteq \exists IsEngagedIn.Activity$$
$$Entity \sqsubseteq \exists IsEngagedDuring.TimeInterval$$
$$Activity \sqsubseteq \forall HasDuration.TimeInterval$$
$$PhysicalSensor \sqsubseteq$$
$$\exists Measure.\left(Temperature \sqcup Pressure \sqcup Noise \sqcup Lighting \sqcup Humidity\right)$$
$$Object \sqsubseteq Picture \sqcup Sculpture \sqcup Cabinet \sqcup Various$$
$$Person \sqsubseteq Visitor \sqcup Employee$$
$$Entity \sqsubseteq \forall isLocatedIn.IndoorLocation$$
$$Entity \sqsubseteq \exists isInterestedIn.Entity$$
$$Room \sqsubseteq \exists hasUsers.Persons$$
$$Room \sqsubseteq$$
$$\forall\left(hasNumberUsers \sqcup hasID \sqcup hasMaxUser \sqcup hasAvailability \sqcup hasFloor\right).Integer$$
$$InPosition \sqsubseteq \forall\left(hasX \sqcup hasY \sqcup hasZ\right).Real$$
$$InPosition \sqsubseteq \exists hasTime.TimeInterval$$
$$IndoorLocation \sqsubseteq \exists hasEnvCondition.EnvCondition$$
$$EnvCondition \sqsubseteq$$
$$\forall\left(hasUmidity \sqcup hasLighting \sqcup hasNoise \sqcup hasPressure \sqcup hasTemperature\right).Real$$

Fig. 3 T-Box for Context Model

In order to model the possible actions in a given environment, the concept of **Activity** is defined, specialized in **Scheduled Activity** and **Deduced Activity**: the first identifies the basic activities, that can be directly obtained by the data outputted by the sensors, and the latter indicates more complex activities, that can be obtained by the application of inference rules on both the sensed data and the information about the entities of interest.

In our model, the deduced activities are those related to user actions. In particular we are able to recognize that a user is watching a picture or a user is getting out of a room and other kind of situations related to environmental parameters (too much persons in a room, temperature of a room greater than a fixed threshold and so on).

For equipping the sensed data with temporal information, the **TimeInterval** Concept is defined, allowing the definition of the temporal duration of the activities. Roles specialized for the domain of Cultural Heritage are defined, as *openInterval*, which identifies the time of opening of the museum, *preCloseInterval*, a variable time just prior to closing, and *closeInterval*, the time in which the structure is closed to the public.

Once the T-Box is being defined, the model is populated by the proper instances, in order to outline the related A-box. The formalized information of the T-Box is then codified in OWL [27], while the A-Box entries are codified in RDF [28]. The Model of the context information is depicted in fig.4.

4.2 The Recommendation Process

The aim of the recommendation process is to select and rank a set of browsing activities. Here, we focus our attention on the recommendation of paintings to watch in an indoor museum, thus we need a recommender system able to suggest the most likely images of paintings for a user.

An effective recommender system for supporting intelligent browsing of multimedia collections has the capability of reliably identify the objects that are most likely to satisfy the interests of a user at any given point of her exploration. In our case, we have to address, more in details, four fundamental questions:

1. How can we select a set of objects from the collection that are good candidates for recommendation?
2. How can we rank the set of candidates?
3. How can we capture, represent and manage semantics related to multimedia objects to reduce the semantic gap between what user is watching and what he is looking for?
4. How can we take into account such semantics in the recommendation process?

To give an answer to the first question we have used the context information. In other terms, when it is recognized that a user is watching a picture in a given room, we select as set of candidates the pictures that are more close to the current one by a *nearest neighbor search* (pre-filtering strategy) based on user location.

To give an answer to the second questions, we have based our recommendation algorithm on an importance ranking method that strongly resembles the *PageRank* ranking system [30] and model recommendation as a social choice problem, proposing a method that computes customized recommendations by originally combining several features of multimedia objects (low-level and semantics), past behavior of individual users and overall behavior of the entire community of users. Our basic idea is to assume that when an object o_i is chosen after an object o_j in the same browsing session, this event means that o_j “is voting” for o_i. Similarly, the fact that an object o_i is very similar to o_j can also be interpreted as o_j “recommending” o_i (and viceversa).

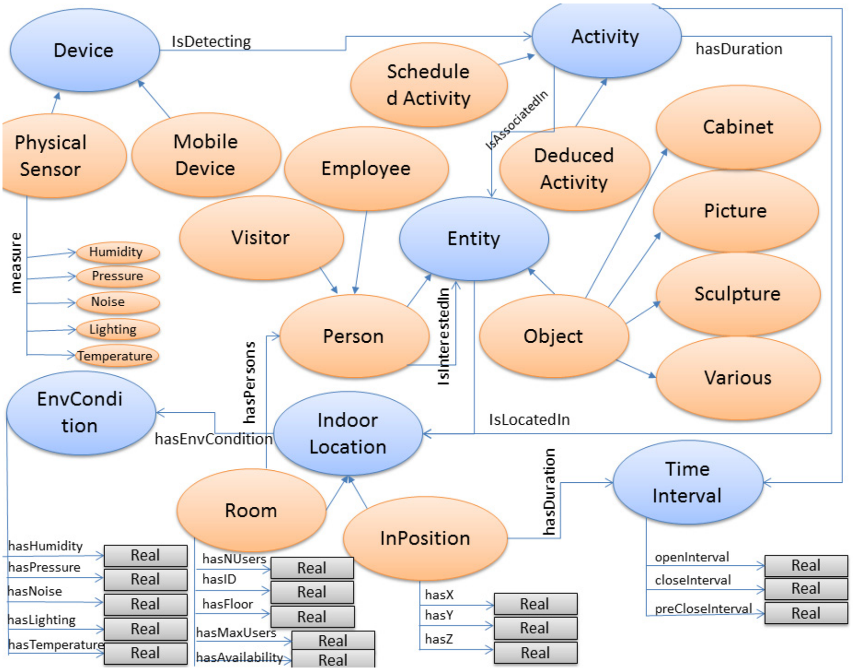


Fig. 4 The Context Model

Thus, our idea is to model a browsing system for a set of object O as a labeled graph (G,l), where $G=(O,E)$ is a directed graph and $l: E \rightarrow \{pattern, sim\} \times R^+$ is a function that associates each edge in $E \subseteq O \times O$ with a pair (t,w), where t is the type of the edge which can assume two enumerative values (*pattern* and *similarity*) and w is the weight of the edge. According to this model, we can list two different cases:

- a *pattern label* for an edge (o_j,o_i) denotes the fact that an object o_i was accessed immediately after an object o_j and, in this case, the weight w^i_j is the number of times o_i was accessed immediately after o_j ;
- the *similarity label* for an edge (o_j,o_i) denotes the fact that an object o_i is similar to o_j and, in this case, the weight w^i_j is the similarity between o_j and o_i.

Thus, a link from o_j to o_i indicates that part of the importance of o_j is transferred to o_i. Given a labeled graph (G,l), we can formulate the definition of recommendation grade of a multimedia object more formally as follows.

*Definition 3.1: (**Recommendation Grade ρ**)*

$$\forall o_i \in O \qquad \rho(o_i)=\textstyle\sum_{oj\in PG(oi)} w^i_j\, o_j \tag{1}$$

where $P_G=\{o_j\in O|(o_j,o_i)\in E\}$ is the set of predecessors of o_i in G, and w^i_j is the normalized weight of the edge from o_j to o_i. For each $o_j\in O$, $\sum_{oi\in SG(oj)} w^i_j=1$ must hold, where $S_G=\{o_i\in O|(o_j,o_i)\in E\}$ is the set of successors of o_j in G.

It is easy to see that the vector $R = [\rho(o_i)\ldots\rho(o_n)]^T$ can be computed as the solution to the following equation:

$$R = C \cdot R \tag{2}$$

where $C=\{w^i_j\}$ is an ad-hoc matrix that defines how the importance of each object is transferred to other objects and can be seen as a linear combination of the following elements [30]:

- A *local browsing matrix* $A_l=\{a^l_{ij}\}$ for each user $u_l\in U$. Its generic element a^l_{ij} is defined as the ratio of the number of times object o_i has been accessed by user u_l immediately after o_j to the number of times any object in O has been accessed by u_l immediately after o_j.
- A *global browsing matrix* $A=\{a_{ij}\}$. Its generic element a_{ij} is defined as the ratio of the number of times object o_i has been accessed by any user immediately after o_j to the number of times any object in O has been accessed immediately after o_j.
- A *multimedia similarity matrix* $B=\{b_{ij}\}$ such that $b_{ij}=\sigma(o_i,o_j)/\Gamma$ if $\sigma(o_i,o_j)\geq\tau$ $\forall i\neq j$, 0 otherwise. σ is any similarity function defined over O which calculates for each couple of objects their multimedia relatedness in terms of low (features) and high level (semantics) descriptors; τ is a threshold and Γ is a normalization factors which guarantees that $\sum_i b_{ij}=1$.

The introduction of matrix B allows to address the two last questions that we introduced at the beginning of the section and thus to introduce a sort of content-based image retrieval with high-level semantics in the recommendation process. In particular, to compute B matrix, we have decided to adopt 5 sets of the most diffused multimedia features (Tamura descriptors, MPEG-7 color-based descriptors, MPEG-7 edge-based descriptors, MPEG-7 color layout- based descriptors and all MPEG7 descriptors [31] and the related similarity metrics have been implemented by LIRE tool. In addition, we exploit specific image metadata (*artist*, *genre* and *subject*) and the semantic similarity has been computed used the most diffused metrics for semantic relatedness of concepts based on a vocabulary [32]. In particular the semantic similarity combines similarities among artists, genres and subjects obtained by using a fixed taxonomy produced by domain experts with image features.

In [30] the experimental protocol to determine the best combination of the proposed metrics is reported for images representing artistic paintings. In particular, the combination between high and low level descriptors is Sugeno fuzzy integral of Li and MPEG-7 color layout- based similarities in order to have higher values of precision, and Sugeno fuzzy integral of Wu-Palmer and MPEG-7 color based similarities in order to have higher values of recall, thus we use this combination for matrix *B* computation.

So far we have a suitable manner to represent object features and to compare the related similarity also considering semantics in terms of object metadata; now, our main goal is to compute customized rankings for each individual user.

In this case, we can then rewrite equation 2 considering the ranking for each user as follows:

$$R_l = C \cdot R_l \tag{3}$$

where $R_l=[\rho(o_i)\ldots\rho(o_n)]^T$ is the vector of recommendation grades, customized for a user u_l.

We note that solving equation 3 corresponds to find the stationary vector of *C*, i.e., the eigenvector with eigenvalue equal to 1. In [29] it has been proved that *C*, under certain assumptions and transformations, is a real square matrix having positive elements, with a unique largest real eigenvalue and that the corresponding eigenvector has strictly positive components. In such conditions, equation 3 can be solved using the *Power Method* algorithm.

It is important to note that *C* takes into account the user's context and does not have to be computed for all the database objects: it need to be computed only for those objects that are good candidates to recommendation.

Finally, the list of suggested items is not fixed and it is arranged on the base of environmental situations. The recommendation degree of objects, that come from rooms with a certain number of persons or with particular values of temperature or humidity, could be penalized and the objects could be excluded from recommendation (post-filtering strategy).

Thus, we use a *memory-based* algorithm so that low and high level similarities are evaluated once; this reflect the unchanging nature of these measures while, clearly, if we add new paintings, similarity matrices have to be conveniently updated. Instead, to capture the dynamic nature of user's behaviour, we periodically re-compute connection matrices; specifically, each connection matrix is updated as soon the browsing session ends.

To solve the cold start problem, when there is no information about user's behaviour, our system uses low or/and high level similarities, in addition to the extracted behaviour of the whole community. For new items, of course, recommendation is based just on similarities.

5 System Customization in the Cultural Heritage Domain

In this section, we describe a case study in the cultural heritage domain for a recommendation system that provides browsing facilities for a multimedia collection of paintings related to an indoor museum. In particular, our recommender helps the users in finding paintings of interest from a large set of choices, proposing a set of suggestions for each observed object; the recommendations are computed combining the of user's behaviour with low and high level image descriptors and considering context information, following the previously described recommendation approach.

The *Knowledge Base* component consists of an image collection of different digital reproductions of paintings (managed by *PostegreSQL* DBMS), to which it is possible to associate artists and artistic genres and the museum cartography (managed by *PostGIS* spatial database).

Each painting can be also linked to a list of subjects, chosen among a list containing the available ones; such information roughly represents what the painting represents. The *Information Integrator*, realized by apposite *JAVA* libraries, allows querying the Knowledge Base and retrieving the desired information.

For what context information sources concerns, we only consider users' mobile devices and a wireless sensor network. By means of apposite *JAVA* libraries and exploiting *TinyDB* [33] and *WIPS* [34] facilities, we have realized two kinds of *Context Providers* able to capture user location and some environmental parameters (temperature, humidity, etc…). The *Context Synthesizer*, realized by apposite *JAVA* libraries, allows us to map the observed context in an OWL ontology and to perform some reasoning activities using *JENA* [35]. Instead, the *Context History* and *Context Analyzer* components have been implemented using *MONDRIAN* OLAP Server[36].

As the *Activity Deliverer* regards, a user can interact with our system using an *Android* application. The presentation logic is based on apposite widgets allowing interacting with users using advanced graphical functionalities. The client requests are elaborated by *JAVA Servlets* and results are sent to the client in form of XML data (according to the *Service Oriented Architecture* paradigm).

As soon as a user interacts with the system and the effective location is computed, the core process starts defining the set of candidates for the recommendation by considering the union of the:

1. the set of paintings which are the most similar to the current one, according the similarity matrices;
2. the set of paintings which are the closest to the current one; if the user is logged in and there exits the related user connection matrix, the past user's behaviour is considered; otherwise the global connection matrix is taken into account.

Eventually, a browsing path is generated considering the cartography of the museum and exploiting the Dijkstra algorithm.

From the final users' perspective, the client application has the following features:

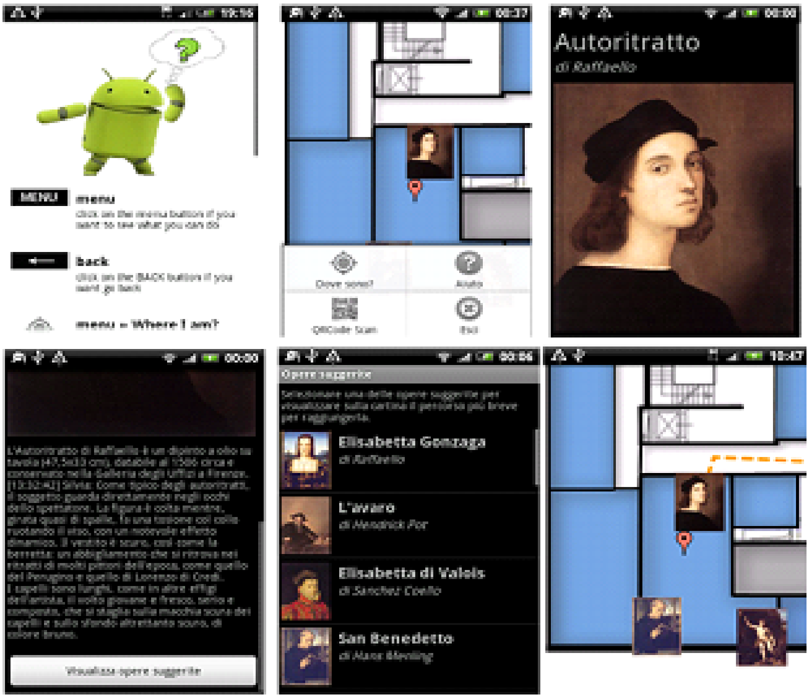
- a set of forms to provide users log in or registration;
- a gallery to visualize images which are returned after a search by author, subject or artistic genre;

- visualization of an image and of the related information and multimedia presentation of recommended images;
- visualization of the browsing paths;
- storing of user session with the information related to the browsing patterns.

Some screenshots related to the client application realized for browsing of a real museum are reported in Figure 5.

6 Preliminary Experimental Results

Recommender systems are complex applications that are based on a combination of several models, algorithms and heuristics. This complexity makes evaluation efforts very difficult and results are hardly generalizable, which is apparent in the literature about recommender evaluation [26]. Previous research work on recommender system evaluation has mainly focused on algorithm *accuracy*, especially objective prediction accuracy. More recently, researchers began examining issues related to users' subjective opinions and developing additional criteria to evaluate recommender systems.



click on the menu button if you want to see what you can do

click on the BACK button if you want go back

L'Autoritratto di Raffaello è un dipinto a olio su tavola (47,5x33 cm), databile al 1506 circa e conservato nella Galleria degli Uffizi a Firenze. [13:32:42] Silvia: Come tipico degli autoritratti, il soggetto guarda direttamente negli occhi dello spettatore. La figura è colta mentre, girata quasi di spalle, fa una torsione col collo ruotando il viso, con un notevole effetto dinamico. Il vestito è scuro, così come la berretta: un abbigliamento che si ritrova nei ritratti di molti pittori dell'epoca, come quello del Perugino e quello di Lorenzo di Credi. I capelli sono lunghi, come in altre effigi dell'artista, il volto giovane e fresco, serio e composto, che si staglia sulla macchia scura dei capelli e sullo sfondo altrettanto scuro, di colore bruno.

Selezionare una delle opere suggerite per visualizzare sulla cartina il percorso più breve per raggiungerla.

Fig. 5 Screenshots of the client application

In particular, they suggest that user's satisfaction does not always (or, at least, not only) correlate with the overall recommender's accuracy and evaluation frameworks for measuring the perceived qualities of a recommender and for predicting user's behavioural intentions as a result of these qualities should be taken into account.

In [30] we proposed a user-centric evaluation of our recommendation strategy, where context data have not been considered in assisting users while exploring a virtual museum. Moreover, we have reported some preliminary experimental results about user satisfaction in using a recommendation strategy for browsing the *Uffizi Gallery* (containing 474 paintings).

The goal of the proposed experimentation was to establish how helpful our system was to provide an exploration of digital reproductions of paintings. Moreover from these experiments we wanted to understand how helpful recommendations offered by our recommender system were to address users toward paintings which satisfied their interests.

In particular, it has been demonstrated that the introduction of recommendation techniques can improve the system usability with respect to assigned browsing tasks and we evaluated such an improvement in terms of empirical measurements of access complexity and *TLX* factors (w.r.t. a system that does not exploit recommendation. i.e. Picasa) provided by different kinds of users.

Specifically, TLX [37] is a multidimensional rating procedure that provides an overall workload score based on a weighted average of ratings on six sub-scales: mental demand, physical demand, temporal demand, own performance, effort and frustration (lower TLX scores are better).

Here, we want to repeat the proposed experimental protocol in a real indoor scenario. To these aims, we have reproduced in our Department a part of the Uffizi Gallery (using reproductions on paper of the paintings) and used a Wireless Sensor Network managed by *TinyDB* to monitor temperature and humidity values of rooms and a *WIPS* to locate a user in a room.

6.1 User Satisfaction

In order to evaluate the impact of the system on the users we have conducted the following experiments. We asked a group of about 20 people (all medium experts in art) to visit the collection of images in our Department and complete several browsing tasks (20 tasks per user) of different complexity (five tasks for each complexity level) without the help of our system. After this test, we asked them to visit once again the same collection with the assistance of our recommender system and complete other 20 tasks of the same complexity.

We have subdivided browsing tasks in the following four broad categories:

- *Low Complexity* tasks (*Q*1)—e.g. "explore at least 10 paintings of *Baroque* style and depicting a *religious* subject";

- *Medium Complexity* tasks (*Q*2)—e.g. “explore at least 20 paintings of *Baroque* authors that have *nature* as their subject”;
- *High Complexity* tasks (*Q*3)—e.g. “explore at least 30 paintings of *Baroque* authors with subject *nature* and with a predominance of *red colour*”;
- *Very High Complexity* tasks (*Q*4)—e.g. “explore at least 50 paintings of *Baroque* authors depicting a *religious* subject with a predominance of *red colour*”.

Note that the complexity of a task depends on several factors: the number of objects to explore, the type of desired features (either low or high-level), and the number of constraints (genre, author, subject). A simple strategy has been used to evaluate the results of this experiment: empirical measurements of access complexity in terms of *number of rooms* and *time*.

In particular, we measured the following parameters:

– *Access Time (t_a)*. The average time spent by the users to visit all the objects for a given class of tasks;
– *Number of Rooms (n_r)*. The average number of visited rooms necessary to visit all the requested objects for a given class of tasks.

Table 1 reports the average values of t_a and n_r, for both without and with the help of our system, for each of the four task complexity levels defined earlier.

Table 1 Comparison between our system and without its assistance in terms of t_a and n_c

Task class	Strategy	t_a (min)	n_r
Q1	Without any help	5	6
Q1	Our System	4.3	5
Q2	Without any help	10.6	10
Q2	Our System	7.6	8
Q3	Without any help	23.2	14
Q3	Our System	16.8	9
Q4	Without any help	32.4	18
Q4	Our System	19.7	11

As showed by the obtained results, our system provides a useful guide to assist users while visiting the museum, improving the user experience by decreasing the requested efforts to complete the assigned task.

We can note that the system performances are very satisfying for the most complex tasks. However, additional improvements can be obtained by exploiting user profiles in order to address users toward paintings which really satisfied their interests, without limit them to visit a predefined set of paintings.

7 Conclusions and Future Work

In this paper we proposed a novel approach to recommendation for multimedia browsing systems, based on a method that computes customized recommendations by combing in an original way intrinsic features (semantic contents and low-level features) of the objects, past behaviour of individual users and behaviour of the users' community as a whole. In particular, we realized a recommender system which helps users to browse reproductions of Uffizi paintings, providing them suggestions computed by our novel method for recommendations. Then we investigated the effectiveness of the proposed approach in the considered scenario, based on the browsing effectiveness and users satisfaction. Experimental results showed that our approach is promising and encourages further research in this direction.

Future works will be devoted to: (i) introduce explicit user profiling mechanism based on the creation of users' categories, (ii) extend experimentation on a larger image data set, (iii) compare our algorithm with respect to other recommendation strategies.

References

[1] The phone of the future. The Economist (December 2006)

[2] O'Brien, J.M.: The race to create a 'smart' google. Fortune Magazine (November 2006)

[3] The Internet of Things. Executive Summary. ITU Internet Reports (November 2005)

[4] Ricci, et al.: Recommender Systems Handbook. Springer (2011)

[5] Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)

[6] Adomavicius, et al.: Incorporating contextual information in recommender systems using a multidimensional approach. TOIS 23(1) (2005)

[7] Kim, H.K., Kim, J.K., Ryu, Y.U.: Personalized recommendation over a customer network for ubiquitous shopping. IEEE Transaction on Services Computing 2(2), 140–151 (2009)

[8] Lam, X.N., Vu, T., Le, T.D., Duong, A.D.: Addressing cold-start problem in recommendation systems. In: Proceedings of the 2nd International ACM Conference on Ubiquitous Information Management and Communication, pp. 208–211 (2008)

[9] Maidel, V., Shoval, P., Shapira, B., Taieb-Maimon, M.: Evaluation of an ontology-content based filtering method for a personalized newspaper. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 91–98 (2008)

[10] Hijikata, Y., Iwahama, K., NishidaI, S.: Content-based music filtering system with editable user profile. In: Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 1050–1057 (2006)

[11] Kazienko, P., Musial, K.: Recommendation framework for online social networks. In: Last, M., Szczepaniak, P.S., Volkovich, Z., Kandel, A. (eds.) Advances Web Intelligence and Data Mining. SCI, vol. 23, pp. 111–120. Springer, Heidelberg (2006)

[12] Manzato, M.G., Goularte, R.: Supporting multimedia recommender systems with peer-level annotations. In: Symposium on Multimedia and the Web (2009)

[13] Baloian, N.A., Galdames, P., Collazos, C.A., Guerrero, L.A.: A Model for a Collaborative Recommender System for Multimedia Learning Material. In: de Vreede, G.-J., Guerrero, L.A., Marín Raventós, G. (eds.) CRIWG 2004. LNCS, vol. 3198, pp. 281–288. Springer, Heidelberg (2004)

[14] Su, J.W., Yeh, H.H.: Music Recommendation Using Content and Context Information Mining. IEEE Intelligent Systems 25(1), 16–26 (2010)

[15] Knijnenburg, B., Meesters, L., Marrow, P., Bouwhuis, D.: User-Centric Evaluation Framework for Multimedia Recommender Systems. In: Daras, P., Ibarra, O.M. (eds.) UCMedia 2009. LNICST, vol. 40, pp. 366–369. Springer, Heidelberg (2010)

[16] Lekakos, G., Caravelas, P.: A hybrid approach for movie recommendation. Multimedia Tools and Applications 36(1-2), 55–70 (2008)

[17] Albanese, M., Chianese, A., d'Acierno, A., Moscato, V., Picariello, A.: A multimedia recommender integrating object features and user behavior. Multimedia Tools Applications 50(3), 563–585 (2010a)

[18] Bazire, M., Brézillon, P.: Understanding Context Before Using It. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 29–40. Springer, Heidelberg (2005)

[19] Shi, et al.: Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In: Challenge on Context-aware Movie Recommendation (2010)

[20] Panniello, et al.: Experimental comparison of pre-vs. Post-filtering approaches in content-aware recommender systems. RecSys (2009)

[21] Oku, et al.: Context-aware SVM for dependent information recommendation. In: Int. Conference on Mobile Data Management (2006)

[22] Ienco, et al.: Parameter-Less Co-Clustering for Star-Structured Heterogeneous Data. Data Min. Knowl. Discov. (2012)

[23] Schifanella, et al.: On context-aware co-clustering with metadata support. J. Intell. Inf. Syst. 38(1) (2012)

[24] Adomavicius, G., Tuzhilin, A.: User profiling in personalization applications through rule discovery and validation. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 377–381. ACM Publishing (1999)

[25] Fawcett, T., Provost, F.: Combining data mining and machine learning for effective user user profiling. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 377–381 (1996)

[26] Schulz, A.G., Hahsler, M.: Evaluation of Recommender Algorithms for an Internet Information Broker based on Simple Association Rules and on the Repeat-Buying Theory. In: Fourth WebKDD Workshop: Web Mining for Usage Patterns & User Profiles, pp. 100–114 (2002)

[27] Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. In: Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications (PerCom 2004), pp. 18–22 (2004)

[28] Lassila, O., Swick, R.R., et al.: Resource description framework (RDF) model and syntax specification. Citeseer Online Publication (1998)

[29] Albanese, M., d'Acierno, A., Moscato, V., Persia, F., Picariello, A.: Modeling recommendation as a social choice problem. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 329–332 (2010b)

[30] Albanese, M., d'Acierno, A., Moscato, V., Persia, F., Picariello, A.: A Multimedia Semantic Recommender System for Cultural Heritage Applications. In: Proceedings of the Fifth IEEE Conference on Semantic Computing – Semantic Multimedia Management Workshop (2011) (to appear)

[31] Lux, M., Chatizichristofis, A.: LIRE: Lucene Image REtrieval - an extensible java cbir library. In: Proceedings of the 16th ACM International Conference on Multimedia, pp. 1085–1088 (2008)

[32] Budanitsky, A., Hirst, G.: Semantic distance in Wordnet: An experimental, application oriented evaluation of five measures. In: Proceedings of the Workshop on WordNet and other Lexical Resources (2001)

[33] Hong, W., Madden, S.R., Franklin, M.J., Hellerstein, J.M.: TinyDB: An acquisitional query processing system for sensor networks. ACM Transactions on Database Systems (TODS) Vol 30(1) (2005)

[34] Kitasuka, T., Nakanishi, T., Fukuda, A.: Wireless lan based indoor positioning system wips and its simulation. In: 2003 IEEE Pacific Rim Conference Proceedings of Communications, Computers and Signal Processing, PACRIM, vol. 1, pp. 272–275. IEEE Publisher (2003)

[35] Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, pp. 74–83. ACM (2004)

[36] Thomsen, C., Pedersen, T.B.: A Survey of Open Source Tools for Business Intelligence. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 74–84. Springer, Heidelberg (2005)

[37] Moroney, W.F., Biers, D.W., Eggemeier, F.T., Mitchell, J.A.: A comparison of two scoring procedures with the NASA task load index in a simulated flight task. In: Proceedings of the IEEE 1992 National Aerospace and Electronics Conference, NAECON 1992, pp. 734–740. IEEE Publishing (1992)

From Patient Information Services to Patient Guidance Services-The iCare Approach

D. Tektonidis, A. Bokma, E. Kaldoudi, and Adamantios Koumpis

Abstract. The provision of the Health services in the EU despite the evolvement of ICT follows a rather traditional path were the patient is totally dependable from his/hers doctors seeking guidance for every decision he/she needs to take related to his/her condition. The vision of the iCare approach is to provide better support to patients from the comfort of their home. This paper presents a new innovative approach to improve Patient Guidance Services (PGS). iCare approach takes full advantage of Semantic Web technologies and IoT and provides a new approach that would put the demands of the patient in the center and exploiting the available sources it will offer patient guidance services reducing dramatically the patient dependency from his/her doctors.

1 Introduction

Health services across the European Union and beyond are increasingly under pressure to deliver better services with diminishing resources. Patients, quite rightly, demand a high standard of service and increasingly also convenience focused on patient needs rather than the ability of the health services to deliver. Patients, on the whole, also prefer to be supported from home, as much as possible, rather than having to travel unnecessarily to receive these services or worse

D. Tektonidis
Research Programmes Division, M.Kalou 6, GR – 54629, Thessaloniki, Greece
e-mail: dte@altec.gr

A. Bokma
Centre for Electronic Commerce, University of Sunderland, Sunderland,
SR6 0DD, United Kingdom
e-mail: albert.bokma@gmail.com

E. Kaldoudi
Democritus University of Thrace, School of Medicine, Alexandroupoli, Greece
e-mail: kaldoudi@med.duth.gr

Adamantios Koumpis
Research Programmes Division, M.Kalou 6, GR – 54629, Thessaloniki, Greece
e-mail: akou@altec.gr

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 413–431.
DOI: 10.1007/978-3-642-34952-2_17

still ending up in long-term care, which are neither desirable nor ultimately affordable.

People nowadays live longer and want to stay in their homes as long as they can. Given this situation there are significant amounts of patients with long-term conditions with risk factors such as cardio-vascular conditions, pulmonary conditions or diabetes to name but a few, who could more effectively be supported at home.

The vision of the iCare is to provide better support to patients from the comfort of their home by providing:

- **Advisory services** to help patients manage their conditions better.
- **Monitoring services** that advise patients when they need to consult healthcare professionals and collect historic data on the patient's condition to devise more informed and better treatment plans.
- **Alerting emergency services** when the patient's condition suddenly deteriorates and urgent help is needed.
- **Dependable services** that will check whether services are functioning correctly and notify patients and service provider when the connection to the patient is lost unexpectedly.

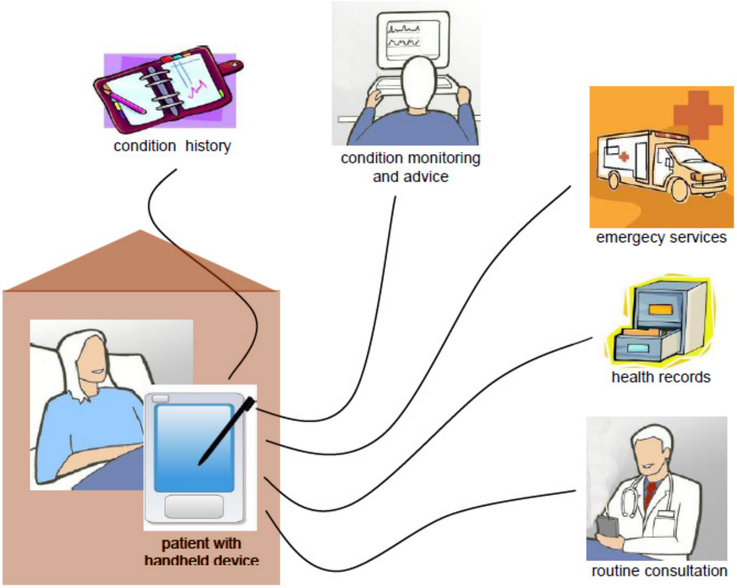


Fig. 1 iCare Patient Guidance Vision

The concept is built on the fundamental principle that patients should be supported as much as possible by easy to use technology to maintain independence through monitoring and advice and involve healthcare services when needed to address emerging problems. Nowadays IoT (Internet of Things) offers new channels and means of communication than enables our approach to be applied to additional user groups that do not have any technological background.

These monitoring and advisory services can be provided by third-party providers (and even include wireless connection to devices such as blood pressure monitors, flow meters, sugar level monitors and the like) to ensure that conditions stay within acceptable limits and offer lifestyle advice to help ease risks. This should include integration with healthcare records to provide healthcare professionals with history information for consultation and treatment. Once consultations are needed these could also be provided phone-consultation if appropriate and healthcare services be alerted when conditions escalate and patients require urgent attention.

As systems are increasingly under attack from hacking and other forms of interference it is essential that the integrity of such services can be assured, checked and that communications and data are treated as strictly confidential. Consequently, service integrity, security and appropriate data access control measures need to be part of the vision.

The iCare approach believes that making this vision reality will have a significant impact on patient care for both patients and healthcare services which go beyond what is currently available in an open service model that allows new service providers to enter and which can easily be extended to different member states independent of their current state of healthcare information technology used.

2 Semantic Interoperability for e-Health

Traditionally, healthcare professionals believed that they knew what was best for the patient [1]. In recent years another view has arisen: that patients are (and should be regarded as) the main experts on their own bodies, symptoms and situation, and this knowledge is necessary for a successful treatment. The patient should thus be treated as a partner in healthcare with both rights and responsibilities [2]. In addition, healthcare politicians and governments might hope that active patients will manage self-care better, thereby easing the economic constraints on the healthcare sector.

As early as 1977 the World Health Organisation advocated that patients participate in their healthcare [3]. Since then, there has been a focus on different ways of strengthening the patients' position in healthcare and influence over medical and treatment decisions [4].

The concepts of patient-centeredness and patient empowerment have been launched in connection with this movement, and offer opportunities for patients to increase their autonomy and involvement in decision making care and treatment [5].

These concepts are widely used and discussed in healthcare research literature, and yet they are rooted in different disciplines and ideologies. Patient-centred medicine was introduced as "another way of medical thinking" by Michael and Enid Balint in 1969 when they proposed to hold seminars on psychological problems in general medical practice [6]. This way of thinking demanded of doctors to include everything they knew about their patient and their understanding their patient as a unique human being before forming an “overall” diagnosis of the patient's illness. In this manner, it can be said that the concept of patient-centeredness originated in a psychological/psychotherapeutic framework. Since then, the concept has been supported as good medicine, yet poorly understood [7].

In contrast to patient-centeredness, the concept of empowerment did not evolve within the healthcare arena, but as a reaction to oppression and inequality within society at large. The roots of the empowerment concept can be traced back to Freire and the “pedagogy of the oppressed” [8] and philosophers like Hegel, Habermas and Sartre or critical social theory and Marxism In the context of critical social theory it involves citizen power and achievement of common goals among people [9]. Women's liberation, gay rights, disability rights and black power were all influenced by empowerment in one way or another. Within the field of healthcare, the concept of empowerment has been used on two levels. First, it has been used to describe a relationship between health and power, based on the assumption that individuals who are empowered are healthier than those who are not [10]. Secondly, it has been used to describe a certain type of patient; one who may become empowered via health education programmes initiated by healthcare systems, or one who may become empowered via their interactions with healthcare providers.

Patient-centeredness and patient empowerment are complementary concepts which do not oppose one-another. Patient empowerment can be achieved by patient-centeredness, but patients can also empower themselves. In any case, all these are mostly realized with a wealth of patient centered guidance services currently emerging in the market.

3 Healthcare System Integration issues and State of the Art

3.1 *Current State-of-the-Art in Integration with Personal Care Devices*

Medical devices are essential to the practice of modern healthcare services. In addition to the hospitals and specialized care units, medical devices are becoming being used for remote healthcare monitoring with the latest advances in wireless communication technology. However, despite the fact that healthcare systems are becoming more dependent on specialized medical devices, the integration of these devices makes existing communication problems more complex.

In addition to the medical device connectivity problem, it is also crucial to provide seamless communication of medical device data into existing healthcare information systems – without this all non-institutional care monitoring will remain in a disconnected information silo. Similar to the extension, IHE has demonstrated this interoperation using the HL7 approved ISO/IEEE 11073 semantic payload with highly constrained HL7 protocols for the communication of vital signs observations and general sensor information. That work was based on another 11073 draft work and was demonstrated successfully in February 2007. Other work is still underway in IHE Patient Care Device Integration Profiles which aims to offer additional levels of integration using ISO/IEEE 11073 common 'MDC' language that healthcare professionals and vendors may use in communicating requirements for the integration of products.

3.2 Current State-of-the-Art in Security and Privacy of Citizen Context and Electronic Health Records (EHRs)

The security and privacy of citizen's context information and EHRs depends on different concepts to be considered such as identity management, authorization, access control, trust and privacy. These concepts are already active research and development areas especially in the eBusiness domain. The major concern in eBusiness applications is the privacy of the customers. In addition, in the healthcare domain the problem is extended to providing the privacy of the records to be accessed.

For the federated identity management, in 1999, Microsoft introduced Microsoft Passport system which provides single sign-on for web sites. Then, in 2001, Liberty Alliance Project was initiated which broadens the focus of identity management with attribute federation and identity provisioning between more than one service providers. Microsoft has also initiated "TrustBridge" project in 2002 in order to provide federation in identity management however not much development has been achieved until now. OASIS Security Services Committee has published Security Assertion Markup Language (SAML) V2.0 with the contributions of Liberty Alliance and Shibboleth initiatives.

Related to the authorization and access control, OASIS Extensible Access Control Markup Language (XACML) standard and IBM Enterprise Authorization Language (EPAL) are the two major industry specifications. Both EPAL and XACML share an abstract model for policy enforcement defined by the IETF and ISO. XACML provides more features like combining result of multiple policies, ability to reference other policies, ability to return separate results for each node when access to a hierarchical resource (fine-grained access control), and support for attribute values that are instances of XML schema elements which are needed for constructing complex policies.

Today the healthcare sector is still using paper based consents usually within a single organisation with very limited patient control. For EHR sharing, the networked health information systems or individual healthcare enterprises mostly use opt-in/opt-out model which either deny the sharing of all records with outside or allows all accesses. The IHE initiative published a profile in 2006, Basic Patient Privacy Consent (BPPC), which provides more choices to patients regarding the sharing of EHR data in IHE document sharing platform. The iCare approach includes an investigation into data access management requirements and suitable technologies and it is expected to contribute to developing a robust mechanism for context and policy sensitive data access management using semantic techniques.

3.3 Semantic Web Technologies

3.3.1 Ontologies and Ontology Engineering

In computer science and information science, an ontology is an explicit specification of a conceptualisation [11] or, more precisely, a formal representation of a set of concepts within a domain and the relationships between those concepts. Both meanings are relevant to iCare, because its common ontology is grounded in a particular philosophical ontology and represented as a computer-/information-science ontology. The iCare approach follows a domain ontology for healthcare services limited to the domain of the approach however in an extensible way to allow redeployment into related areas. The modelling will include not only services but also actors and policies and preferences to help manage the data access and sharing aspects.

Ontology building is supported by several methodologies proposed, in the literature. Some methodologies focus on building ontologies manually without a priori knowledge (e.g., [12]). Others are dedicated to the cooperative building of ontologies. There are also methodologies for reverse-engineering existing ontologies. And finally there are learning ontologies from various sources, such as texts, dictionaries, knowledge bases, relational schemas, XML documents etc. (e.g., [13], [14]). Another practical approach is by using constraints for cleaning initial taxonomies, as exemplified by OntoClean [15]. iCare expects to use the ontoclean approach to verify the well-formedness of the domain ontology it will use.

Ontology population is the process of inserting concept instances and relation instances into an existing ontology without changing its structure. Examples of approaches and practical systems performing ontology are Artequakt [16], the KnowItAll system [17] and SOBA [18].In the most of existing methods, instances are extracted from text which are not directly relevant to iCare, but iCare will be concerned with building bridges between the domain ontology and the underlying data model. In this case, ontology population requires a concept instance extraction toolkit which will be investigated.

Ontology languages are formal languages used to represent ontologies. The Web Ontology Language (OWL, [19], [20]) has quickly become the standard for the worldwide and the semantic web. There are three variants of OWL. OWL Full is compatible with the Resource Description Framework (RDF), but not usable for formal analysis and has different semantics from the other types of OWL. OWL DL is the maximally expressive variant that is also computationally complete. OWL Lite is intended as a lighter weight alternative to OWL DL, but is not much used in practice. Some ontology researchers use "even lighter" variants of OWL with better computational properties because, for realistically-sized problems, even OWL Lite quickly becomes computationally intractable.

Semantic annotation and meta-data is an approach to enrich information sources with additional semantic information, typically by referencing external semantic resources. Uren [21] notes that existing systems for annotating documents provide good user interfaces that are well suited to distributed knowledge sharing and enables the annotation of legacy resources but, at the same time, their support is lacking in degree of automation and range of documents covered, addressing issues of trust, provenance and access rights and resolving the problems of storage, and keeping annotations consistent with evolving documents, particularly in combination with evolving ontologies. Hence, semantic annotation is still an evolving field. There are a number of proposals for semantic annotation (or semantic mark-up) of web services. One type is METEOR-S, SAWSDL and WSDL-S, which annotate information in WSDL with ontologies. Another type is OWL-S, SWSF and WSML, which offer dedicated ontology languages for semantic web services. In [22], semantic annotation of process models is a prerequisite for semantic business process management. Some latest achievements are based on the SUPER project. The SUPER ontology is used for the creation of semantic annotations of both BPMN and EPC process models in order to support automated composition, mediation and execution. However, the annotation mechanism is based on WSMO. Semantic annotations are introduced for validation purposes, i.e., to verify constraints on the process execution semantics. In general, any enterprise model can be annotated for enabling interoperations (INTEROP 2007). iCare is watching developments in this area with interest although it is assumed that due to the safety critical nature of the application a manual intervention or at least a supervised approach may be more appropriate. The need for this may also be substantially reduced if standard case adapters are used for service development.

Semantic annotation of web-services is a common approach to support semantic web services, by linking the web-service descriptions to an ontology. For example the micro-WSMO approach [23], describe the various types of service semantics by means of a RDF Schema. Furthermore they use Semantic Annotations for WSDL and XML Schema (SAWSDL) to define a place for a semantic description in a Web service. The result of this process is an extended WSDL with additional semantic annotations that conforms an standard ontology. Another approach starts with web services that are textually specified by HTML documents. For example, SA-REST introduces semantic annotations, which are based on the RDFa ontology language, inside the meta- or container HTML elements (SPAN, DIV etc.) from a web service specification. These annotations characterize the different services, their operations and messages etc. Accordingly, Kopecky et al. [24] define the hRESTS HTML microformat, which represents a REST service functionally using the CALSS and REL HTML elements. This approach also reuses the SA-REST notation for describing the different data formats supported by the REST service. The main advantage of these approaches is that they reuse the existing textual specification and make them machine-readable. Furthermore, since REST services are described by ontologies, they can be mapped further to other format. Ontology matching is also called alignment, matching, matchmaking and mapping. Although some authors make more precise distinctions between them, we will use the term ontology matching here. The main issue in ontology matching is to find semantic links (such as equivalence, disjointness or subsumption) between the concepts and the relations in two distinct ontologies that cover overlapping domains. The methods are based on different strategies: hierarchical clustering techniques, formal concept analysis, analysis of terminological features of concepts and relations (i.e., names or natural-language definitions) or analysis of structure [25]. Much of this issue may be addressed in iCare in the configuration phase of services and the use of a shared domain ontology reduces the need for this.

Rule based engines and semantic transformations support semantic interoperability and application integration by augmenting the ideas behind data transformation languages like XSLT with semantics. Hence, technologies that originally were used for rules definition like RuleML (http://www.ruleml.org/) and SWRL (SWRL http://www.daml.org/2003/11/swrl/) are now used to define semantic transformations. The transformation of data is defined inside transformation ontologies (SWRL) and the transformation engine also uses ontologies that define the semantics of the data. The Rule Interchange Format (RIF) is an attempt to support more complex cases and allowing m-to-n transformations. Finally, Ontorule (http://ontorule-project.eu/) use ontologies to create interoperable business rules. The execution of the semantic transformation is based on rule engines that are able to process RuleML or SWRL. There are many tools that use SWRL files directly, like SweetRules (http://sweetrules.projects.semwebcentral.org), or enhancements on Java rule engines, like JESS (http://www.jessrules.com/). These reasoning engines can be used for semantic transformation in combination with SWRL.

3.4 Semantic Web Services

Service-Oriented Architecture (SOA) allows applications to share common business logic or methods [26]. SOA is used to wrap legacy systems to make their functions and data more readily available within and across organisational boundaries, and to develop new systems on top of which cross-functional workflows can be established as composite services based, e.g., on agent technologies or enactable process models. The heavy focus on services in modern enterprise information architectures has led to the promotion of service-oriented computing (SOC) as a new paradigm for ICT in the private and public sectors (Cummins 2002, Gold-[27]). iCare will contribute to this area to develop a semantically enhanced web service architecture that goes beyond the interface level and focused on an integration-centric approach.

Web services are a central part of the technological platform for SOA/SOC. The W3C defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network. For this purpose, the interfaces that a web service provides must be described in a machine-readable format. One central standard is the web-service description language (WSDL), which describes web services in terms of one or more interfaces defined in terms of their input and output data. Typically, methods are exchanged using the simple object access protocol (SOAP) over TCP/IP. Web services over WSDL/SOAP are further supported by the extensive OASIS-managed “WS”-family of standards for security, privacy, transactions etc. But WSDL/SOAP is not the only platform to support web services. Other examples are regular APIs that are made available on the Internet, as well as OMG's CORBA, Microsoft's DCOM and Java RMIs.

WSDL/SOAP-based web services are becoming criticised for being overly complex. In consequence, the representational state transfer (REST) principles have been proposed as a path to offering web services without the complex platforms and standards of the WSDL/SOAP family.

Although RESTful services are being touted as a light-weight alternative to protocol-heavy web services based on WSDL/SOAP, we will continue to use the term web services in a wide sense in iCare. Hence, we will use the term web services whether they are based on WSDL/SOAP or other protocols and whether they conform to RESTful principles or not.

There are several families of proposed standards for semantic web services. OWL-S uses OWL for describing the semantics of web services that are defined using WSDL, and it is itself an extension of OWL.

The purpose is to enable users and software agents to automatically discover, invoke, compose, and monitor services under specified constraints (W3C 2004). OWL-S offers some support for describing composite semantic web services that are put together from other simpler ones, and there is tool support for executing simpler services specified in OWL-S. The WSMO family is an alternative to OWL-S that is maintained by the ESSI cluster (WSMO 2009). It is not based on OWL, but consists of the web service modeling ontology (WSMO), which is a conceptual model for Semantic Web Services, the web service modeling language (WSML), a language which provides a formal syntax and semantics for WSMO, and the web service modeling execution environment (WSMX), which is an execution environment and a reference implementation for WSMO. WSMX offers support for interacting with semantic web services. Recently, metamodels were defined for two of the three prominent Semantic Web service descriptions languages. Skogan et al. [28] and Guarino et al. [15] describe a metamodel for OWL-S and Skogan et al. [28] discusses a metamodel for WSML. The Platform Independent Metamodel for Semantic Web services (PIM4SWS) can be combined with model transformations to selected individual meta-models of semantic web-service formats (OWL-S, WSML, SAWSDL) to allow transfer of information between platforms. The PIM4SWS in combination with a model-driven Semantic Web services matchmaker agent allows discovering semantic services independent of selected description formats like OWL-S, WSML and SAWSDL (Semantic Annotation of WSDL and XML Schemas). It is expected that iCare builds on the OWL-s approach and extend the standard architecture for deeper annotation of services and their integration and generate innovation in this field alongside the semantic handling of data access request (though that could also be classed separately under the heading of data access management).

4 The iCare PGS Model

What patients need is to keep an eye on their conditions and receive useful advice such as reminding them to take their medication or adapt the dosage to their current situation and also lifestyle advice that may help to ease their condition where appropriate. Patients should also have available advice if they are worried that will take them through their current condition to check whether everything is fine or whether they should seek medical advice soon or immediately. This requires collecting data from patients through dialogue and/or through integrated devices (for example if a blood pressure measuring device is used with wireless connection) and keeping these records for further analysis or for making available during consultations.

To implement this scenario the advisory services need to be available to the patient via a handheld device or if too complex to be run on the device through remote access to such a service hosted elsewhere. This service needs to be integrated with healthcare systems and services as shown in figure 1 to provide the necessary connections to make available the data collected from the patient to clinicians for routine or emergency consultations and to support advisory and alerting services should the patient need assistance.

A considerable amount of technology in terms of patient monitoring is available for clinical use in hospitals and health-centres (such as Micropaq™ from http://www.welchallyn.com) to name but one) but not so much available technology has found its way into the patients home or to enable patients otherwise confined to their home to become more mobile and self-sufficient. There is huge potential to achieve improvements in quality of life while reducing direct contact with healthcare providers through the development of mobile patient monitoring and guidance services which this approach aims to address.

Our approach is based upon an open platform for mobile patient guidance services aimed at patients with long-term risk-factors such as cardio-vascular conditions, diabetes or pulmonary conditions. Mirroring the key objectives stated in the vision outlined above this can be provided through:

- the use of ***pre-diagnostic and advisory services*** the patient can use to check whether their condition is still manageable and perhaps receive also lifestyle guidance to improve their condition or stop them from deteriorating.
- together with the ability to ***collect condition histories from patients for inspection*** by healthcare service staff so that the healthcare service can make improved condition management.
- ***Alerting services*** where needed to trigger healthcare service intervention so that patients who suddenly need help are spotted early and their needs attended to before they become critical.
- The services are aimed at ***helping patients manage their conditions*** and provide practical advice as well as monitoring conditions and making the history available to the healthcare service provider to make more informed decisions and intervene where conditions are suddenly deteriorating.

The benefit of this concept is to enable patients to manage their long-term conditions just as effectively from the comfort of their home and free resources including hospital beds to other patients in need of them.

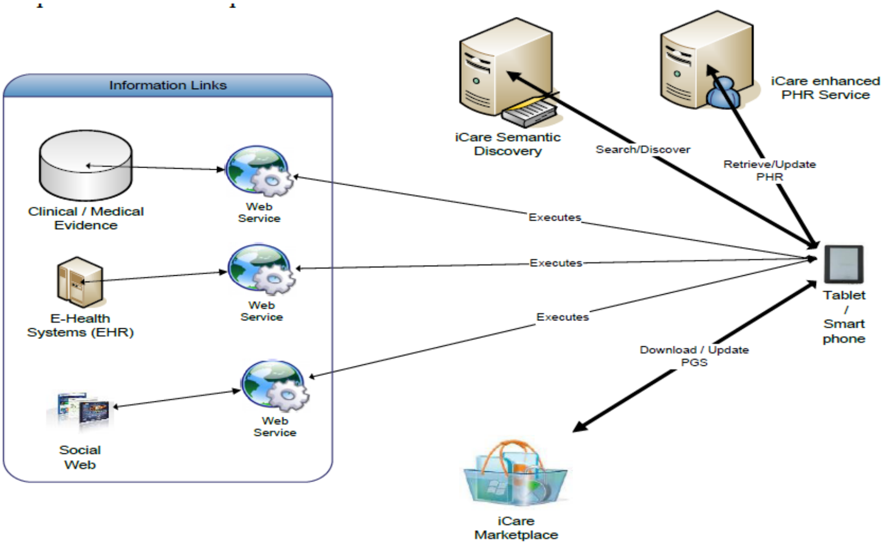


Fig. 2 iCare Patient Guidance Services

From a technical point of view, and as far as the patient is concerned, the iCare service concept is *centred on the use of standard and widespread mobile devices* such as smartphones or tablet computers and to use these to run internet based services from a variety of approved providers alongside existing devices to measure blood sugar level, pulse and blood pressure or lung capacity. The applications would then use dialogue and data entry by patients for *offering advice* and to *keep a history* used and uploading to healthcare information systems for inspection by healthcare professionals during consultation (on or off-line) and to *trigger emergency response*. The diagram below shows the technical service infrastructure where patients can download suitable services and use them to interact with relevant parties and systems to receive advice and support from a variety of approved providers.

Concerning the service concept, the iCare approach acknowledges the fact that there are a *variety of providers* and a potentially *growing number of evolving services* that need to be supported and thus proposes an open platform for service provision for ease of access from the patients and healthcare services provider's perspective. iCare also acknowledges the *sensitivity of the data* associated with treating patients and the need for suitable approaches for *security and privacy* enforcement. Consequently there is a need for an open platform that allows available services to be published and discovered and given the high degree of connectivity required a SOA based service registry is envisaged. This platform is designed to support the following:

- Development of patient guidance service components and their correct classification using a standard iCare service adaptor
- Approval of service components for correct classification and functioning

- Exposing service components and publishing them in the iCare marketplace
- Selecting suitable services for a candidate application
- Downloading client to patient device and Composing a complex service
- İnstantiation and configuration of service at patient home and testing
- Handling service updates

The benefit of this approach is that services can rapidly be selected and configured for multi-end point services as suggested by figure 1. To this end figure 3 shows the service offerings in the iCare marketplace which is published through the iCare Semantic Discovery Registry and that can be used to access the chosen iCare enhanced PHR Service and in addition to connect to several additional services for example to upload the history data to the patient record system or connect to telemedicine services or even emergency services where available.

The approach focuses on services related to the lifestyle of patients with chronic disease and especially with cardiac and renal problems. These two groups have been chosen because these are very indicative cases of patients that require frequent consultations. Therefore a measure of success could be the reduction of the patient-doctor contact as well as the perceived quality of the service and advice from the patients' point of view.

4.1 The iCare PGS Semantic Service Platform

The provision of the services will follow the life-event model that has been implemented successfully in the past for e-Government Services[2]. The life-event approach facilitates discovery of the service because it bases the services discovery in the user profile (in iCare situation this can be the PHR of the patient) and on WHAT the patient wants to do ("My ***blood pressure*** is ***x/y*** – do I need to take more ***medication*** or do I need to see a ***doctor*** or do I need an ***ambulance*** straightaway?"). To increase the reusability and reduce the complexity of a Patent Guidance Service, iCare will divide a PGS into a set of operations/actions that the patient may require. Although these operations will be part of the PGS the will be autonomous enabling the patient to select the operations that would like to use.

As presented in figure 1, the patient will be able to download the Patient Guidance Services from the iCare marketplace according to his/her needs. The services will be installed as Apps (Mobile Applications) to his/her mobile device (Tablet/Smartphone). The installed PGS will use data from the PHR of the patient through the iCare enhanced PHR service, the input provided by the patient and using the iCare Semantic Discovery Service will be able to locate and retrieve information from the available information link. We use the term "information link" as in Linkeddata to define every available system or service that can be accessed through the web.

The iCare Marketplace will provide a variety of patient guidance service applications that can be linked to the healthcare service and data from the PHR will be

[2] OneStopGov project `http://islab.uom.gr/onestopgov/`

able to support patients with chronic disease reducing dramatically the dependency from their carers and improving their lifestyle.

4.2 PGS and Data Collection as Interoperability Components

iCare defines Patient Guidance Service and data collection as a standalone component that conceptually is very close to the notion of mobile applications. The major issue in using a PGS is if the service can be applied to applications or systems that the patient uses. The variety of available software currently used in healthcare and of communication APIs create a very large number of different scenarios for PGS.

In iCare the usage of a service breaks down into components that are independent but which will be interoperable. To facilitate the development of such components, iCare will provide for healthcare application providers semantic descriptions of the PGS or data collection services. Following the approach of Linkeddata[3] where software systems are defined as Information Links, iCare will use Semantic Web Technologies to define the requirements of interoperability components and data access and security constraints.

It is essential that services are appropriate for the patient and that data in heterogeneous environments is handled correctly. It is also essential that appropriate safeguards about data access, privacy and security are obeyed in a context-sensitive way. This can only be achieved through the use of semantic approaches and more specifically the semantic web as we are dealing here with a web-based platform. These requirements are consequently expressed in high level semantics that define what information should be exchanged and what operation should be performed before or after the exchange. Therefore the specifications of the Interoperability Component can be defined without any dependencies from the software applications that will participate. An Information Link can participate in the operations of a PGS as long as there is a formal way to associate the semantics of the integration case (vocabulary and operations) to the integration protocol used by the Information Link.

Therefore there is a need for a semantic annotation between the semantics of the Integration Case and the semantics of the Information Link. Since the operations and the vocabulary of an Integration Case is independent from the Information Links the development of interoperability components for different Information Links is reduced to the semantic annotation and the technical mapping between the concepts and the actual data.

This facilitates the creation of Interoperability Components Marketplace that will contain predefined Integration Cases and will enable Software Development companies and Integration Companies to develop and publish interoperability components to enable interoperability of their software or software that they support. The interoperability components will be based on the semantics of the Integration Case and the semantics of the Information Link and be made available as a basis for a given application.

[3] `http://www.linkeddata.org`

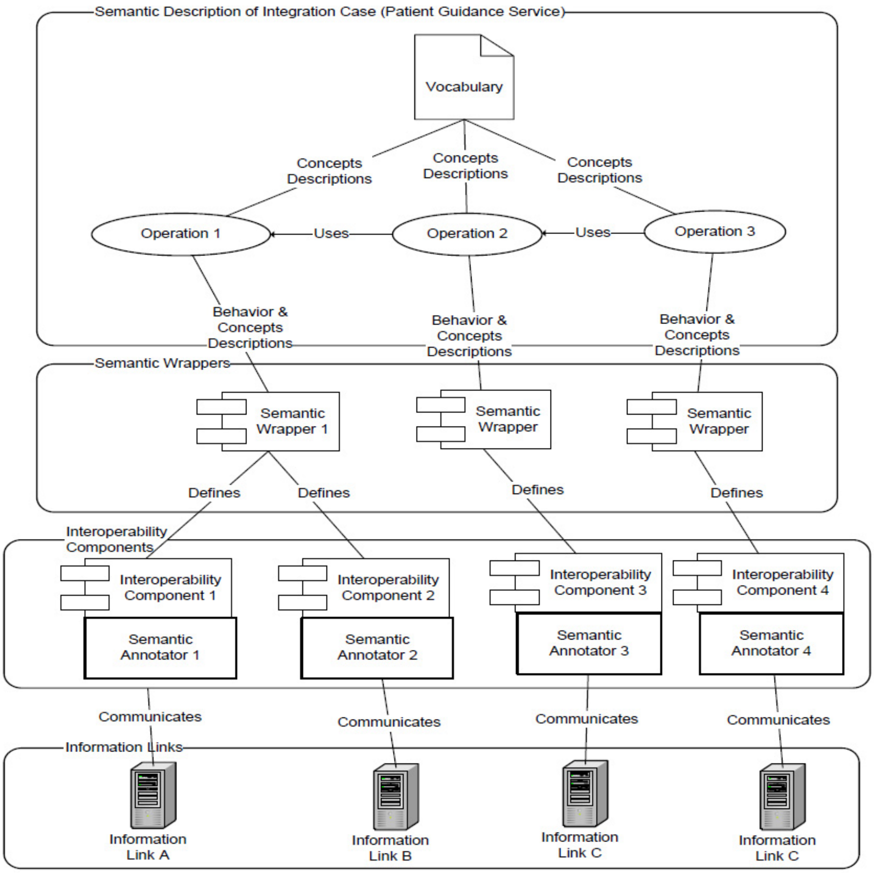


Fig. 3 iCare Patient Guidance Services Deployment Approach

In figure 3, the Patient Guidance Service is broken down into operations that are defined by a vocabulary. The vocabulary used for the definition of the service will use Semantic Web technologies such as ontologies and it will also use standards and vocabularies used in healthcare systems. For example if the PGS involves a medicine definition SNOMED[4] may be used. As aforementioned a PGS will be defined as a set of operations that the patient will be able to perform. The operation will be deployed as a Semantic Wrapper that is the software implementation of the operation. The Semantic Wrapper is a stand-alone software component that implements the behaviour and the vocabulary of the operation. The semantic wrappers can be used be smart devices so that these devices to provide and receive data from iCare.

[4] http://www.ihtsdo.org/snomed-ct/

Finally, in order a Semantic Wrapper to become an interoperability component that can be executed it requires annotation that will enable to communicate with an information link. The information link can be a software application or a web service in general that the interoperability component will interoperate. The semantic annotator will provide a bridge between the interoperability component and the service or the system that will be implemented for.

5 A New Approach on Patient Guidance Services Provision-iCare Innovation and Novelty

iCare presents a new approach for the provision of Patient Guidance Services. The patient can “buy” the Services from a Marketplace (along with additional components) as an App and use it from his/her mobile device. The approach also focuses on the personalization of the services and the secure and reliable sharing of the patient data amongst the doctors that are involved in their treatment. Therefore the novelty of the approach can be summarised as follows:

- **Interoperability Components Marketplace:** For iCare both the usage of PGS and the data collection software are mobile Apps that can be downloaded from a Marketplace similar to iTunes. From the patient perspective this enables the patient to search for the PGS that are suitable to his/her occasion (treatment) and to his environment (hospital or healthcare that he/she visits). From the business perspective, iCare Marketplace want to attract not only software companies that has developed a particular e-health application (e.g. EHR) but also other companies that can develop an interoperable component for an e-health application.
- **Case Based, on-Demand Integration.** The execution of a PGS may involve several e-Health applications. The data collection for executing the services depends on the e-Health applications that are involved. iCare introduces an “integration on demand” where the software components will be selected according to the e-Health of each case. However the semantics of the PGS are independent from the e-Health applications. iCare enable the use of Semantic Web Technologies to define the semantics of the PGS that will provide formal (technical) conceptualization that will facilitate the development of the interoperability components.
- **Mobility and Stand-Alone Functionality of the PHR and PGS Mobile Applications.** The usage of mobile device that will contain the patient data (part of the PHR) instead of using a typical online system such as Google Health or Microsoft Vault is the availability of the data. The precondition for using online PHR system is the existence of the internet connection and a computer that are not always available. iCare enables both online and offline repositories to ensure that the patient will have available his patient record at all time. The portable device will be able to connect directly to an e-Health application (hospital EHR) or device. In addition, the patient will be able use his current location data that are important to may PGS especially if the patient travels frequently.

- **Semantically Enhanced Integration and Data Protection:** The use of semantic web technology for context sensitive integration of services and the parties involved will lead to a higher degree of reliability of integrations and services fit for purpose. Current techniques are too much focused on the parameters of the interface rather than the purpose and use of the service they provide access to. In addition, the implementation of a context sensitive access control mechanism will deliver services which implement stringent data access policies.

5.1 The iCare Solution

The iCare aims to provide a complete solution to the new vision for PGS provision presented previously. In this solution the patient plays of course the main role however we believe that the success of the platform lays also to the support provided by the Platform to 3rd party companies even to SMEs that would like to include new PGS or enrich the existing ones.

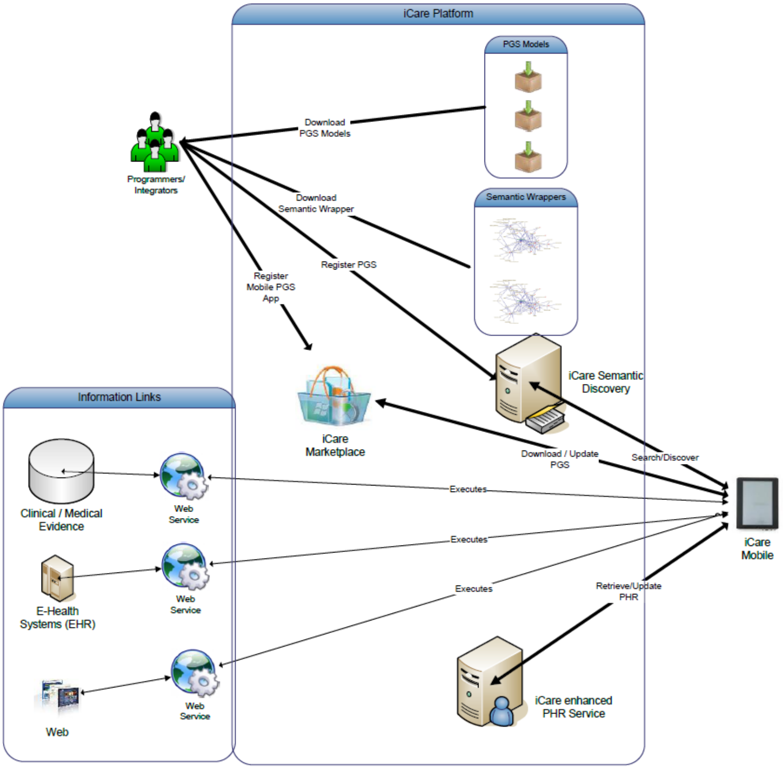


Fig. 4 The iCare Solution

The iCare solution (figure 4) aims to create and support a community of developer that will be able to communicate with the end customers (patients) through a marketplace similar to the concept of iTunes. In this respect the companies that aim to create PGS will be assisted with several tools and resources.

Therefore the iCare solution consists of the following components:

- **PGS Models.** The PGS model are predefined generic models of PGSs that can be used for the developement of PGS. A PGS model contains the semantics (vocabulary) and the functionality that provides a technically defined desciption for the developer.
- **Semantic Wrappers.** The PGS will use sources from the web to acquire their information. The iCare will provide a pool of semantic wrappers that enable the annotation of a PGS to various information links. Therefore the developer needs only to technically "map" the Semantic Wrapper to the PGS in order to use the information links.
- **iCare Semantic Discovery (Search Facility).** The semantic Search facility enables the patient to search using semantic queries available PGS. The facility also enables the developers to register their PGSs defining their functionality.
- **iCare Enhanced PHR Service.** iCare aims to automate the execution of the PGSs. Therefore the iCare enhanced PHR Service will communicate with iCare Mobile in order to provide only the data required for the execution of the PGS. Therefore the patient will not need to input information that already exists in his/her PHR.
- **iCare Marketplace.** The Marketplace contains all the necessary functionality to facilitate the download and installation (and update) of PGS mobile apps. iCare focuses on the simplicity of the usage of the marketplace.
- **iCare Mobile.** The iCare Mobile enables the patient to search, download and use PGS from his/her mobile. In addition, it contains functionality that requires data from his/her PHR through the iCare enhanced PHR Service.

iCare solution aims to simplify the execution of PGS. The entire solution is built upon principals that have been already applied very successfully in other domains (like mobile applications). However we strongly believe that with the usage of Semantic Web the concept of Apps Marketplace can be applied to PGSs providing an innovative approach to PGS provision.

6 Conclusions and Future Work

The concept of Patient Guidance Services presented by the EC aims to enhance Patient Information Services integrating information from the EHR of the patients and including sophisticated decision support services. The knowledge should be created from sources available from the Web. In this direction iCare is an approach that is based on a modular architecture that exploits Semantic Web technologies and SOA to build a system focuses on patient with chronic diseases.

The next phase of the development of the iCare approach is the integration of the main components and a pilot operation that will enable us to assess the presented architecture. The pilots will be implemented to renal and cardiac patients including advisory and alerting services before investigating more complex services.

References

1. Emanuel, E.J., Emanuel, L.L.: Four models of the physician–patient relationship. J Am. Med. Assoc. 267, 2221–2226 (1992)
2. Coulter, Paternalism or partnership? Patients have grown up-and there's no going back. Br. Med. J. 319, 719–720 (1999)
3. Bissell, P., May, C.R., Noyce, P.R.: From compliance to concordance: barriers to accomplishing a re-framed model of health care interactions. Soc. Sci. Med. 58, 851–862 (2004)
4. Ong, L.M., Visser, M.R., Lammes, F.B., de Haes, J.C.: Doctor–patient communication and cancer patients' quality of life and satisfaction. Patient Educ. Couns. 41, 145–156 (2000)
5. Little, P., Everitt, H., Williamson, I., Warner, G., Moore, M., Gould, C., et al.: Preferences of patients for patient centred approach to consultation in primary care: observational study. Br. Med. J. 322, 468–472 (2001)
6. Balint, E.: The possibilities of patient-centered medicine. J. R Coll. Gen. Pract. 17, 269–276 (1969)
7. de Haes, H., Koedoot, N.: Patient centered decision making in palliative cancer treatment: a world of paradoxes. Patient Educ. Couns. 50, 43–49 (2003)
8. Crawford Shearer, N.B., Reed, P.G.: Empowerment: reformulation of a non-Rogerian concept. Nurs. Sci. Q 17, 253–259 (2004)
9. O'Cathain, A., Goode, J., Luff, D., Strangleman, T., Hanlon, G., Greatbatch, D.: Does NHS direct empower patients? Soc. Sci. Med. 61, 1761–1771 (2005)
10. Roberts, K.J.: Patient empowerment in the United States: a critical commentary. Health Expect. 2, 82–92 (1999)
11. Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies 43, 907–928 (1992)
12. Gruninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI 1995 (1995)
13. Maedche, A., Staab, S.: Ontology Learning for the Semantic Web. IEEE Intelligent Systems 16(2), 72–79 (2001)
14. Babko-Malaya, O., Romero, M., Kallmeyer, L.: LTAG - Semantics for questions. In: Proceedings of TAG, Vancouver, pp. 186–193 (2004)
15. Guarino, N., Welty, C.: Evaluating Ontological Decisions with OntoClean. Communications of the ACM 45(2), 61–66 (2002)
16. Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., Shadbolt, N.R.: Automatic Ontology-Based Knowledge Extraction from Web Documents. IEEE Intelligent Systems 18(1), 14–21 (2003)

17. Etzioni, O., Kok, S., Soderland, S., Cagarella, M., Popescu, A.M., Weld, D.S., Downey, Shaker, T., Yates, A.: Unsupervised named-entity extraction from the Web: An experimental Study. Artificial Intelligence 165, 91–134 (2005)
18. Navigli, R., Velardi, P.: Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 27(7), 1063–1074 (2005)
19. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: Staab, I.S., Studer, R. (eds.) Handbook on Ontologies in Information Systems. Springer (2003)
20. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. J. of Web Semantics 1(1), 7–26 (2003)
21. Uren, V., Philipp, C., Jose, I., Siegfried, H., Maria, V.-V., Enrico, M., Fabio, C.: Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. Journal of Web Semantics 4(1), 14–28 (2006)
22. Wetzstein, B., Ma, Z., Filipowska, A., Kaczmarek, M., Bhiri, S., Losada, S., Lopez-Cobo, J., Cicurel, L.: Semantic, Business Process Management: A Lifecycle Based, Requirements Analysis. In: Workshop SBPM, Innsbruck, Austria, June 7, pp. 1–11 (2007)
23. Vitvar, T., Kopecký, J., Viskova, J., Fensel, D.: WSMO-Lite Annotations for Web Services. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 674–689. Springer, Heidelberg (2008)
24. Kopecky, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: semantic annotations for WSDL and XML schema. IEEE Internet Computing 11(6), 60–67 (2007)
25. Shvaiko, P., Euzenat, J., Leger, A., McGuinness, D.L., Wache, H. (eds.): Contexts and Ontologies: Theory, Practice and Applications (C&O-2005) Proceedings of the AAAI 2005 Workshop C&O-2005. AAAI Press (2005)
26. Linthicum, D.: Next Generation Application Integration: From Simple Information to Web Services. Addison-Wesley, USA (2003) ISBN: 0201844567
27. Bernstein, B., Ruh, W.: Enterprise Integration. Addison Wisley (2005) ISBN 0-321-22390-X
28. Skogan, D., Grounmo, R., Solheim, I.: Web Service Composition in UML. In: The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC), Monterey, California (September 2004)

Fingerprint and Iris Based Authentication in Inter-cooperative Emerging e-Infrastructures

Vincenzo Conti, Salvatore Vitabile, Luca Agnello, and Filippo Sorbello

Abstract. E-infrastructures must support the development of heterogeneous applications for workstation network, for mobile and portable systems and devices. In this context and relating to all collaborative and pervasive computational technology a very important role is played by security and authentication systems, which represent the first step of the whole process. Biometric authentication systems represent a valid alternative to conventional authentication systems providing robust procedures for user authentication. On the other hand, Internet of Things involves a heterogeneous set of interacting devices to enable innovative global and local applications and services for users. In this chapter fingerprint and iris based unimodal and multimodal authentication systems will be described, analyzed and compared. Finally, a prototyped embedded multimodal biometric sensor will be outlined. Software and hardware prototypes have been checked against common and widely used databases.

Keywords: Biometric Authentication Systems, Unimodal and Multimodal Systems, Embedded Sensors.

1 Introduction

The e-infrastructures are becoming more widespread and pervasive and, by enabling effective sharing of information and coordination of activities between diverse, dispersed groups, they are expected to transform knowledge-based tasks [1]. E-infrastructures must support the development of heterogeneous applications

Vincenzo Conti
Facoltà di Ingegneria e Architettura e delle Scienze Motorie
Cittadella Universitaria, Enna, 94100, Italia
e-mail: vincenzo.conti@unikore.it

Salvatore Vitabile
Dipartimento di Biopatologia e Biotecnologie Mediche e Forensi
via del Vespro, Palermo, 90127, Italia
e-mail: salvatore.vitabile@unipa.it

Luca Agnello · Filippo Sorbello
Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica
viale delle Scienze Palermo, 90128, Italia
e-mail: {luca.agnello, filippo.sorbello}@unipa.it

N. Bessis et al. (Eds.): *Internet of Things & Inter-cooperative Comput. Technol.*, SCI 460, pp. 433–462.
DOI: 10.1007/978-3-642-34952-2_18

for workstation network, for mobile and portable devices, which are not necessarily inter-operable and inter-cooperative for effective data portability, service and resource sharing, discovery, scheduling and integration. Specifically, the rapid developments in networking and resource integration domains have resulted various distributed and collaborative computational technologies including Web 2.0, social networking, SOA, P2P, sensors, Grids, Clouds and Crowds.

In this context and relating to all collaborative and pervasive computer technology, a very important role is played by security systems. Among them, authentication systems represent the first step in the whole process. Conventional authentication systems (based on username and password) are not able to guarantee a suitable security level for several applications. With more details, the security requirements must ensure secure and trusted user information to protect sensitive data resource access and they could be used for user traceability inside the platform. Biometric authentication systems represent a valid alternative to the conventional authentication systems [2] providing a flexible e-infrastructure towards an integrated solution supporting the requirement for improved inter-organizational functionality. Among biometric traits, two strong and invasive features, such as fingerprint and iris, have been addressed and analyzed considering both unimodal and multimodal biometric authentication systems.

In the chapter, fingerprint and iris based unimodal and multimodal approaches will be described, analyzed and compared. Finally, a prototyped embedded biometric sensor for inter-cooperative emerging e-infrastructures will be outlined and compared as well [3], [4], [6]. However, software developed biometric authentication systems could undergo several types of attacks, such as Replay Attacks, Communication Attacks, and Database Attacks [8]. Embedded biometric sensors could be a solution to exceed the security limits of the conventional software recognition systems, hiding the most common attack points of a biometric authentication system [7]. An embedded biometric sensor is composed of a biometric scanner for traits acquisition and a hardware processing core. The use of FPGA technology for systems prototyping leads to an acceptable accuracy, great potential speedup, and interesting power consumption feature [5], [6].

The chapter is organized as follows. Section 2 describes the Internet of Things. Section 3 introduces the analysis of fingerprint recognition. Section 4 describes the fingerprints pre-processing and features extraction algorithms. Section 5 introduces the analysis of iris recognition. Section 6 describes the iris pre-processing and features extraction algorithms. Section 7 depicts the biometric features fusion techniques. Section 8 proposes the embedded biometric sensors. Section 9 outlines the experimental results on official databases. In Section 10 some useful discussions and comparisons with the state-of-the-art approaches are provided. Finally, Section 11 reports the conclusions.

2 Internet of Things

Internet of Things (IoT) is a new interesting and a technological increasing paradigm. Innovative global and local applications and services for users will exploit the paradigm changing fruition and enabling modes. IoT will enable places, people, and 'things' (physical objects) connectivity, bringing together distributed and collaborative computational technologies including Web 2.0, social networking, SOA, P2P, sensors, Grids, Clouds and Crowds. IoT will necessarily consist of a heterogeneous set of devices and heterogeneous communication strategies between the devices and it will require interoperability at multiple levels.

The 'things' on the IoT are various physical entities and different technologies for connecting them to the IoT are (or will be) used. The major classes of physical devices considered in the scope of this chapter are: (i) attached devices: identifiers such as RFID tags or bar codes are attached to things to enable their automatic identification and tracking; (ii) sensing and actuating devices: they are placed in the close vicinity of the 'things' and provide an alternative access to their properties or functions; (iii) embedded devices: 'things' like industrial machinery, home electronics, smart phones, wearable devices have embedded processors, data storages, sensors and actuators for its functionality and capabilities.

As reported in [36], IoT resources could be conceived as service end-points for Machine-to-Machine (M2M) platforms, i.e. Things/Resources as a Service (TaaS or RaaS). Consequently, a solid, well-designed M2M/IoT platform should provide the basis for the simplified management of resources. These solutions work as huge collaborative and interoperable network for the transmission of end-terminals, such as sensors and smart objects, gathered data to the backend systems, such as servers, which runs applications or data processing solutions. It's clear that one of the most important problems to approach is related to the security context: every resource and user in the IoT context should be authorized to act and communicate.

The aim of this chapter is to present a set of techniques for biometric user identification in inter-cooperative emerging e-infrastructures. In the above context, the security of 'thing' is one of the main issues and a simple code as well as the (username, password) pair seems too lite to enable security mechanisms for applications and services confidentiality and integrity. On the other hand, a prototyping strategy for embedded biometric sensor development is proposed. The authentication 'thing' is an embedded sensor able to acquire and compare biometric traits for user authentication. The hardware device allows for 'local' biometric traits processing and matching without sensible data transmission during the authentication task. The use of heterogeneous devices should evolve into a more structured set of solutions, where 'things' communicate with other entities and they are closely integrated with Internet hosts, infrastructure, and services.

3 Fingerprint Recognition

Fingerprints are characterized by a series of ridges; between two adjacent ridges there is a valley. Careful analysis of image acquisition can find other peculiarities, distinguished in macro and micro-features.

3.1 Macro-features and Micro-features Definition

Macro-features fingerprint [11] are generally used for image classification, and they can be divided into:

- *Singularities*: are regions where the ridges have a particular trend (pronounced curvatures, loops, confluences, whorls, etc...) and they belong to three distinct classes: loop, delta junction and whorl. In these areas the "drawing" of the crests is in the shape of an O, Δ, U;
- *Flow lines*: hypothetical lines ideally parallel to a group of contiguous ridges; since not being real lines, the flow lines are not precisely defined;
- *Directional image*: is a matrix of the directions obtained by overlapping a grid with a series of nodes and thinking of drawing in each node a vector with a parallel direction to the flow line passing through the node itself;
- *Ridge count*: is a parameter that indicates how the ridges are close in a certain region and is represented by the number of ridges intersecting a segment having as its extreme two hypothetical points on the fingerprint.

On the contrary, in a fingerprint there are local discontinuities along the ridges: terminal points and bifurcation points called minutiae [11]. The type of discontinuity in the lines determines the type of minutia. Each minutia is uniquely identified by:

- *Type*: terminal point and bifurcation point;
- *Location*: it is represented by minutiae coordinates in the Cartesian system representation;
- *Orientation*: it represents the angle formed by the vector which determines the direction, and the direction of the minutia with the horizontal axis.

3.2 Matching Techniques

Matching is the process that establishes the similarity degree between two fingerprints using the above described features. The main difficulties in the matching phase are due to changes of factors such as translation, rotation and the epidermis that can be different during each acquisition. The most common methods for fingerprints matching proposed in the literature are:

- *Matching based on the correlation*: it consists in "overlapping" two images in order to calculate the difference of the corresponding pixels. This type of comparison requires a phase of fingerprint image alignment (image registration);
- *Matching based on minutiae*: it is the most common used technique. Minutiae are extracted from the fingerprint image and stored as a set of points. The comparison consists in finding the maximum coincidence between the same types of minutiae in corresponding positions between the online acquired and stored fingerprint.

4 Fingerprints Pre-processing and Features Extraction Algorithms

The feature extraction process consists of a series of pre-processing algorithms on original RAW data. Pre-processing phase generally consists of:

- *Normalization*: it allows standardizing pixel intensity, so they can have value in a fixed range;
- *Gabor filtering*: a Gabor function [16] is realized by modulating a sine wave or a co-sinusoid with a Gaussian function in order to obtain spatial and frequential information. In fact, the image decomposition is carried out through a pair of quadrature Gabor filters, the 2D Gabor wavelet (see Figure 1a);
- *Segmentation*: it consists in separating foreground and background regions; foreground regions contain ridges and valleys of the fingerprint. Generally, the background regions are characterized by a very low level of variance in contrast to the foreground that has high variance, for this reason the variance threshold method is used [17] (see Figure 1b);
- *Thinning*: in order to reduce the ridge thickness to a single pixel (1 pixel-wide), the Zhang-Suen thinning algorithm [18] can be used; the image is binarized and the algorithm is iteratively applied until no pixel is a candidate for elimination (see Figure 1c). This step is necessary to minutiae localization.

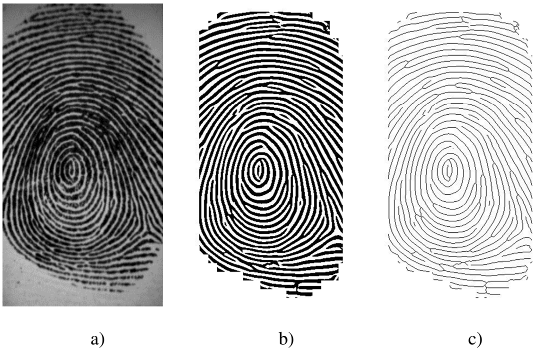


Fig. 1 a) Original fingerprint; b) image enhanced by Gabor and Segmentation algorithms; c) fingerprint after thinning algorithm

4.1 Micro-features Extraction

Each ridge pixel is categorized according to the number of transitions 0->1 calculated using a 3x3 window in a clockwise direction (see Figure 2a, 2b). The border minutiae are removed using a mask (see Figure 2c and Figure 2d).

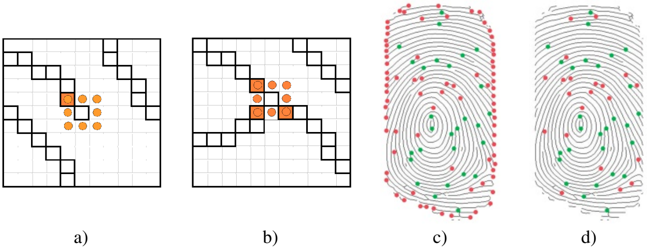


Fig. 2 a) Terminal point; b) Bifurcation point; c, d) Boundary minutiae removal

4.2 Macro-features Extraction

The algorithms used for macro-features extraction are based on the pattern analysis of singularities regions and topological information such as relations between different regions [2]. Macro-features are generally used to classify fingerprints in five macro-categories [8].

Algorithms based on macro-features are characterized by the lowest computational load required for feature extraction. The singular points could not be detected or be corrupted in the image acquisition process, for this reason, the singularity point extraction phase is always associated with a phase of topological analysis within the matching algorithm [19].

The most common method to locate singular points is based on the *Poincaré index* method [17]. The process is divided into several tasks: *Normalization, Segmentation, Orientation* [20], and finally for each value of the directions matrix the *Poincaré index* is calculated (see Figure 3).

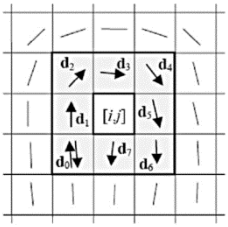


Fig. 3 Poincaré indexes calculation

4.3 Advantages and Limits

The previously analyzed techniques show many differences about computational cost: the systems based on the extraction of micro-features, in fact, needs a phase of pre-processing much more complex than ones based on the macro-features.

Every fingerprint contains many micro-features and they are located across the footprint. There is no fingerprint, complete or partial, that does not contain micro-features. Since many minutiae are present in every fingerprint, they are the features most used for fingerprint recognition: usually 10 minutiae are sufficient to declare the identity of a person.

On the contrary, the macro-feature extraction has a less computational cost, but they don't give a high final accuracy. In a fingerprint there are only 2 types of singularity points, named *core* and *delta*, and sometimes in the acquisition phase we can lose one of them. This leads to an analysis that is not accurate and a less reliable level of recognition. In the literature, in fact, the singularity points are used only for the fingerprint classification.

A choice that can lead to a compromise would be the simultaneous use of micro- and macro-features: a pre-processing phase more light and application of minutiae extraction algorithms only in the identified singularities zones [19].

5 Iris Recognition

The visible part of the iris is divided into two main zones which are often different from the color: the ciliary zone and the area separated from the pupillary internal collars (*hedge*) that has a zigzag pattern structure, called *Collarette* (see Figure 4).

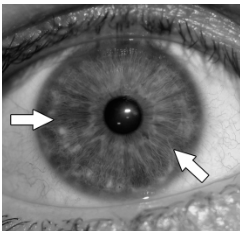


Fig. 4 Collarette is the white contour indicated by arrows

Iris features are random and they aren't dependent by genetic factors (the pigmentation color is the only genetic feature). Moreover, in each person's iris differs from one eye to another. It exhibits about 266 features against 90 of the fingerprints; the iris temporal invariance is guaranteed by the cornea and it isn't subject to diseases that can change its appearance. The probability to find two identical irises is one in 10^78, so the iris is a valid biometric identifier [13].

Iris biometric systems have evolved thanks to recent ophthalmology studies held by Flom and Safir [12], and by the Australian Society of Ophthalmologists [35].

The biometric system phases are the following:

- *Eye image acquisition*: typically it is performed by a CCD camera, that tries to acquire with the maximum definition the human iris;

- *Iris localization*: iris is extracted from acquired images, localizing the portion of the image between external (limbo) border, and internal (pupil);
- *Image normalization*: the Cartesian coordinates system is modified into polar coordinates representation, because the iris area isn't constant but it varies in relationship to pupil expansion (see Figure 5);

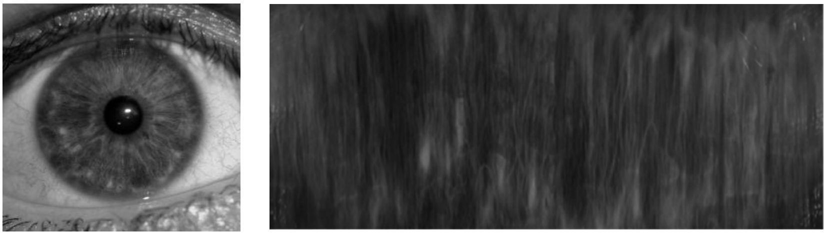


Fig. 5 Iris normalization procedure

- *Features extraction*: micro (*nucleus*, *collarette*, *valleys*, *radiants*) and macro (*frequency code extraction*) features extraction;
- *Coding*: it consists in the pattern extracted construction of micro or macro features;
- *Matching*: it compares an acquired iris against an iris stored in a database, using a metric(i.e. Hamming distance).

5.1 Macro-features and Micro-features Definition

The Gabor filter is widely used in frequency-domain based approaches to obtain and codify localized information. The Log-Gabor filter implementation, proposed by Field [14] can be used. The Log-Gabor filter can be constructed with arbitrary band. It's a Gabor filter constructed as a Gaussian on a logarithmic scale. The filter frequency response is:

$$G(f) = \exp\left(\frac{-\left(\log\left(\frac{f}{f_0}\right)\right)^2}{2(\log\left(\frac{\sigma}{f_0}\right))^2}\right) \tag{1}$$

where f_0 is the center frequency and σ determines the bandwidth of the filter.

In the second method implemented in [14], the iris has been encoded using the Log-Gabor filter. In particular has been used the algorithm written by Libor Masek [15], which consists in considering each row of the normalized image, as a 1D signal that is convolute with the 1D Log-Gabor filter. The output filter is then quantized into four levels. The coding process is illustrated in [15]. The encode function generates a biometric template of normalized iris and a mask that represents noise in the image: this information will be stored in a database and it will represent a user.

On the contrary, iris micro-features of interest are localized using approaches based on textures and they are commonly the following:

- *Nucleus*: points equal to a deformation of the hypothetical circular contour of the pupil;
- *Collarette*: points determining the division of iris in an area close to pupil and sclera, representing a ragged outline;
- *Crypt*: points in the area between the collars and iris-sclera outline corresponding to increased pigmentation;
- *Radial furrow*: points equal to a change in pigmentation having a shape equal to a circumference arc;
- *Freckles*: points within the area identified by the collars, equal to segments going from the contour of the pupil to collars contour;
- *Pigment-spots*: points equal to the crypt but with a reduction in pigmentation.

5.2 Matching Techniques

Iris comparison could be performed using one of the following approaches:

- *Base frequency method*: during segmentation phase the outputs are the sclera-iris and iris-pupil contours, and these features are encoded by applying the Gabor wavelet, which returns a binary code of 256 bytes, called *Iris code*.
- *Texture analysis method*: this category is illustrated from Wildes [21], [22], it analyzes texture in the biometric template. Like Daugman approach, it consists in acquisition, segmentation, normalization, extraction and identification phases. Iris outline is extracted using *Hough* transform;
- *Zero-distance method*: Sanchez and Reillo [23],[24],[25] studied the grayscale tonal variation in the biometric template. This features extraction step is based on a multi-scale zero-crossing representation that gets a computerized model made with 256 bits, called Sanchez-Reillo iris signatures.

6 Iris Pre-processing and Features Extraction

In the following a brief description about algorithms that leads to features extraction:

- *Pupil and iris localization*: in order to find the pupil zone, a binary thresholding is necessary (see Figure 6a);
- *Iris and pupil segmentation*: iris must be detected in the portion just above the upper eyelids and lower eyelids. The segmentation process must be sensitive to different levels of contrasts, robust respect to case of irregular edges and able to operate in conditions of variable occlusion (see Figure 6a);
- *Eyelashes detention*: this phase highlights and removes pixels belonging to the eyelashes. The algorithm presented here is the one used in [26] (see Figure 6b, 6c);
- *Eyelids segmentation*: algorithm searches the set of points describing the boundary iris-eyelid and approximating a parabolic curve (see Figure 6d);

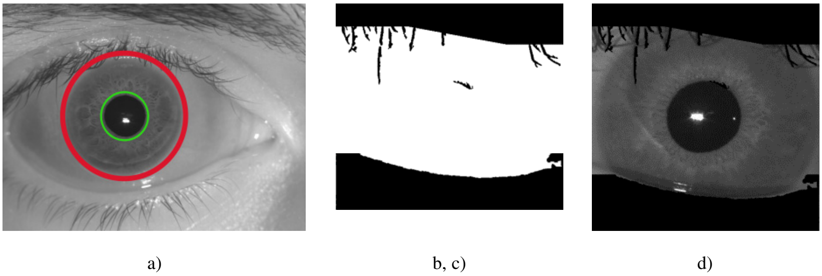


Fig. 6 Iris extraction phases

- *Normalization step*: standardization system used in this work has been proposed by Daugman and called *Rubber Sheet Model* [32]. It does correspond to each point of the iris region a pair of polar coordinates (r, θ) where r belongs to the interval [0,1] and θ to the interval [0, 2π] (see Figure 7).

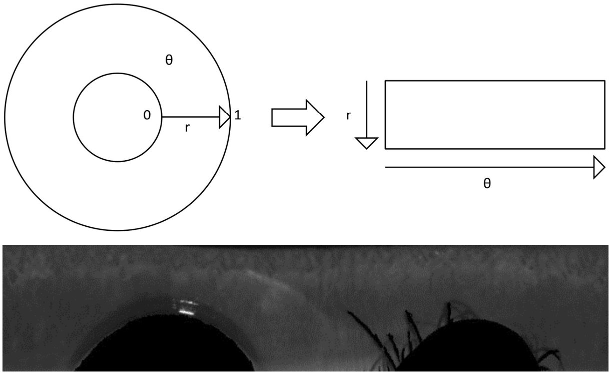


Fig. 7 Coordinate transformation for normalization

Although this system solves the problems of pupil dilation, the acquisition of images at different distances and of non-concentricity of the pupil with respect to the sclera, does not compensate rotation problems. This is done in the phase matching by rotating the iris template until there is alignment with the template stored in memory.

6.1 Micro-features Extraction

The coding is performed in two phases: the micro features extraction and their encoding according to a suitable pattern:

- *Collarette extraction*: this step initially provides a subdivision of the image containing the iris polar circular crown;

- *Nucleus extraction*: a threshold is used in order to obtain the binary image. Successively, this task proceeds with the extraction of the edge allowing to extract the *nucleus* (see Figure 8).

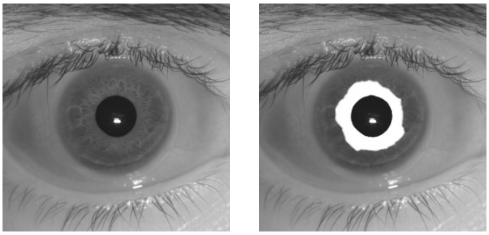


Fig. 8 Collars extraction example

6.2 Macro-features Extraction

After the pre-processing step, a digital representation of the extracted iris is needed to allow for image storing common format. The most common technique based on the application of the Gabor filter and its enhanced version known as the algorithm of Log-Gabor, used both to extract frequency information from analyzed images, will be illustrated. As said before, the algorithm written by Libor Masek has been used [15].

6.3 Advantages and Limits

As for fingerprints, the different feature extraction techniques provide different pre-processing stages. The most critical phase is the image acquisition and the subsequent iris segmentation phase. The first two approaches are based on two different domains: frequential, with regard to Daugman's work (analysis of the base frequencies, macro-features), and spatial, with regard to the works of Wildes (texture analysis, micro-features).

In particular the Daugman one, based on macro-features extraction, the ability to work on the entire portion of iris by applying some transform in frequency domain is a definite advantage, both in terms of computational cost and accuracy. With regard to these approaches, they are quite stable when applied to images of medium/high quality. On the contrary, works based on texture analysis and on micro-features extraction, provide a more accurate extraction and further information at the microscopic level.

7 Fusion Techniques

Multimodal systems can be classified in according to the number of sources and biometric number of "samples" used [27]. The main feature of a multimodal system is to combine different information to arrive at a decision. We can distinguish different types of fusion and consequently different systems architectures [28], [29]. In relation to the decision policy founding fusion algorithms, there are two possible approaches: the pre_mapping fusion strategy applied before the matching phase and the post_mapping fusion strategy applied after the matching phase.

Pre_mapping fusion has two subcategories: sensory data level fusion, in which data coming from sensors are combined before the remaining processing steps; and a feature extraction fusion level, in which the extracted biometric information coming from different modalities are fused before the remaining processing steps. In *post_mapping* fusion, there are two subcategories, as well: a matching score level fusion (also known as level review), in which the results of two independent matching systems are combined with a weighted rule; and the decision level fusion, in which a decision system is used to process unimodal system output decision.

Widely-used *pre_mapping* fusion approaches refer feature set fusion, while widely-used *post_mapping* fusion approaches refer independent matching score fusion.

7.1 Matching Score Fusion Level

In multi-biometric systems based on the matching score fusion level, the feature vector from each sensor are compared with the respective samples (recorded separately) so that each subsystem can give its own opinion, in the form of points (matching score), which indicates how the feature vector is similar to sample [2]. These scores are then combined into a single result which will be forwarded to the supervisor who is responsible of the final decision based on "matching score zone."

In ranked list combination fusion method (which does not require a normalization process), the list produced by each classifier can be interpreted as the opinion of the classifier itself. In this way, this method can also be seen as a fusion to a matching score level (see Figure 9).

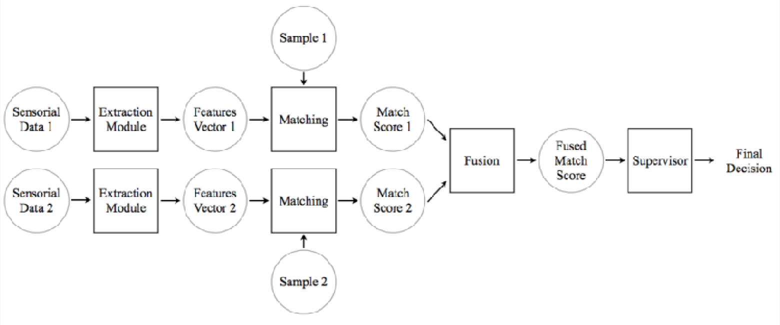


Fig. 9 Matching score fusion level

7.2 *Feature Set Fusion Level*

The information obtained from each biometric system is successively merged into a single vector [2], [30], [31]. Figure 10 shows this kind of fusion.

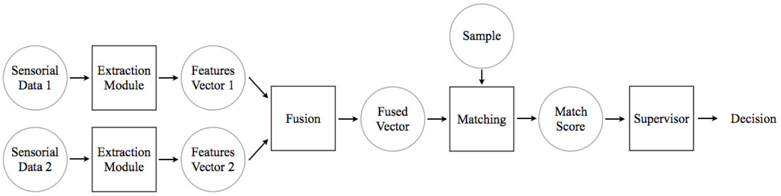


Fig. 10 Extraction level fusion

Two methods to obtain results are the following:

- *Weighted sum*: this method can be used only if the features are commensurate; the fusion performs a weighted sum between the various vectors of extracted features from respective sensors;
- *Concatenation*: it is used in the extraction level fusion where each feature vector is independent from all other (i.e. they are not commensurate as for example a system that works with voice and face). In this case it is possible to concatenate these vectors into a single feature vector. The vector thus obtained will represent the identity of a person in a different (and more discriminating) features space.

The homogeneous vector obtained by data extracted is composed of binary sequences representing unimodal biometric models [2]. A header and a biometric template thus compose the resulting vector.

8 Biometric Embedded Sensors

A biometrics identification system is able to scan and map the biometric features for users, register them in a database, creating a template that can be checked against all further scans to verify the user's identity. A biometric system can be considered trusted if and only if it withstands some typical attacks [7], [8]:

- *Replay attacks*: attacks due to the replication of information processed during the authentication process;
- *Communication attacks*: attacks valued in terms of resistance to the interceptions of the information during its transmission;
- *Database attacks*: attacks due the manipulation of information contained in the database.

A possible solution for overcoming the above issues is the implementation of an embedded sensor, containing biometric templates, implementing the whole processing module with no biometric data transmission before user authentication. In addition, the choice of an embedded device overcomes some limits as system performance and response time, as well as more specific problems as the vulnerability and accessibility of personal information. Biometric sensors could also be integrated by the needed solutions for keys management and distribution, involving decentralized Certification Authorities, as proposed and evaluated in [9], [10].

The embedded biometric sensor exceeds the security issues in the biological data management phase (replay attacks), since biometric traits processing and matching are performed on-board, in the embedded sensor. In addition, the use of cryptography (biological templates could be ciphered using symmetric and/or asymmetric encryption algorithms) avoids clear information transmitting inside the platform (communication attacks). Tamper-resistance devices, such as smartcards, are able to store, protect, and transport user biometric identifiers. Encryption algorithms, such as AES [45], can be adopted for data integrity and protection. With this solution no distributed or centralized database is needed, avoiding the presence of databases linked point of failure (database attacks) [6].

9 Experimental Results

Many tests to verify performance of the techniques and algorithms above described and exposed have been done. Experimental trials have been carried out on three recognition systems:

- Multimodal fingerprint/iris recognition system;
- Multimodal fingerprint/fingerprint recognition system.

In what follows, two multimodal recognition systems are analyzed and described. The first one is a fingerprint/iris multimodal recognition system operating in the frequency domain. Experimental results are referred to a software implementation. The second one is a fingerprint/fingerprint multimodal recognition system operating in spatial domain. Experimental results are referred to a hardware implementation.

9.1 Accuracy Indexes

The measurement indexes used in this work are the well-known False Acceptance Rate (FAR) and False Rejection Rate (FRR)to detect false positives and false negatives respectively. In order to compare the results with those in the literature and have some scientific value, the protocol used to calculate the number of tests and procedures is that of international competition FVC [33]. Even though the competition was created for fingerprints, from a year or two its criteria became a standard for any type of biometric recognition system. This protocol will be used both for the unimodal and multimodal systems. The protocol provides the following practice tests for the determination of:

- *FRR calculation*: each sample/image that contains the biometric features of an individual is compared with the remaining samples/images of the same individual. If, during the comparing step the sample *g* image is compared with the sample *h* image, then the reverse comparison (i.e., *h* against *g*) is not performed to avoid an absolutely obvious and already calculated result;
- *FAR calculation*: the first sample/image belonging to each individual is compared with the first sample/image of all individuals, and remaining in the database. If the sample image *g* corresponds to *h*, the reverse comparison/symmetric (i.e., *h* against *g*) is not performed to avoid an absolutely obvious and already calculated result.

9.2 Dataset Description

The tests have been performed on two official databases freely downloadable from the Internet. As for the system based on fingerprints, the FVC2002 database will be used [33], while as regards the system based on irises the BATH database will be used [34]. From these two databases have been constructed others containing other pairs fingerprint/iris to simulate the acquisition of two biometric features from a generic user. Both used databases are among the most commonly used in science to assess the robustness and performance of recognition systems.

The FVC2002 DB2 Fingerprint Database

The FVC2002 DB2 [33] database containing fingerprints was constructed by acquiring images through an optical sensor. This database is divided into two parts, DB2A and DB2B, and is composed of a total of 880 images (800 within the database DB2A and 80 within the database DB2B) belonging to 110 users. For each user 8 images with a resolution of 296x560 px have been processed. Table 1 summarizes the FVC2002 DB2 database characteristics.

Table 1 FVC2002 DB2

Database	Images	Users	Acquisitions	Resolution (px)
FVC2002 DB2A	800	100	8	296x560
FVC2002 DB2B	80	10	8	296x560

The BATH Iris Database

The BATH database [34] consists of 2000 images belonging to 50 different ethnic groups of users. For each user are considered left and right eye, capturing 20 images for a total of 40 images for the user. Table 2 summarizes the characteristics of BATH database.

Table 2 BATH database

Database	Images	Users	Acquisitions	Resolution (px)
BATH	2000	50	40 (20 for eye)	1280x960

The Multimodal Database: FVC2002 DB2/BATH

To test the proposed fusion technique and the multimodal system various databases were realized; they are reported in Table 3, and whose description is successively reported.

Table 3 FVC2002 DB2/BATH

Database	Images	Users	Acquisitions	Resolution (px)
FVC2002 DB2A	800	100	8	296x560
FVC2002 DB2A-S1	400	50(1-50)	8	296x560
FVC2002 DB2A-S2	400	50(51-100)	8	296x560
FVC2002 DB2B	80	10	8	296x560
BATH	2000	50	40 (20 for eye)	1280x960
BATH-S1	80	10	8(1-8)	1280x960
BATH-S2	400	50	8(1-8)	1280x960
BATH-S3	400	50	8(9-16)	1280x960

In particular:

- the FVC2002 DB2A-S1 database was generated considering the first 50 members of the original database;
- the FVC2002 DB2A-S2 database was generated by considering the last few 50 users of the original database;
- the BATH-S1 database was generated with 10 users extracted in a very casual way from the original full database. For each user, we have considered the first eight acquisitions of the left eye;
- the BATH-S2 database was generated considering 50 members of the original database. For each user was considered the first eight acquisitions of the left eye;

- the BATH-S3 database was generated considering the same 50 members of the original databases but each of them was considered a member of the 8 images acquired in the next 9 to 16.

Table 4 Example databases

Database	Fingerprints Database	Irises Database	Users	Acquisitions (Fingerprint+Iris)
DBtest 1	FVC2002 DB2B	BATH-S1	10	8+8
DBtest 2	FVC2002 DB2A-S1	BATH-S2	50	8+8
DBtest 3	FVC2002 DB2A-S1	BATH-S3	50	8+8
DBtest 4	FVC2002 DB2A-S2	BATH-S2	50	8+8
DBtest 5	FVC2002 DB2A-S2	BATH-S3	50	8+8

Finally, Table 4 shows the databases on which the final results have been reported for both unimodal and multimodal systems and which will be discussed in the next section.

9.3 Iris/Fingerprint Multimodal System

The proposed multimodal biometric system is composed of two main stages: the preprocessing stage and the matching stage (see Figure 11). Iris and fingerprint

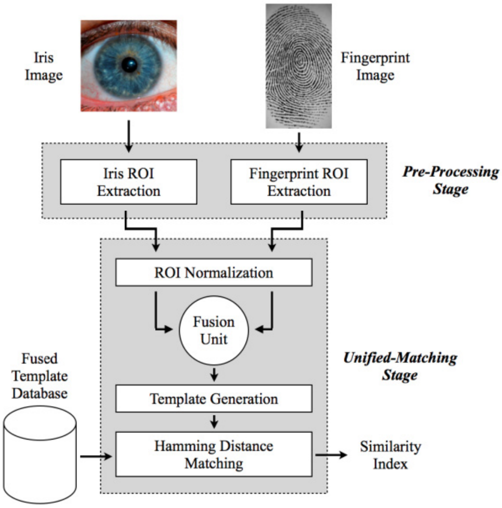


Fig. 11 General schema of the proposed multimodal system

images are preprocessed to extract the regions of interest (ROIs), i.e. singularity regions, surrounding some meaningful points. Despite to the classic minutiae-based approach, the fingerprint-singularity-regions-based approach requires a low execution time, since image analysis is based on a few points (core and delta) rather than 30–50 minutiae. Iris image pre-processing is performed extracting the iris region from eye and deleting eyelids and eyelashes. The extracted ROIs are used as input for the matching stage. They are normalized, and then, processed through a frequency-based approach, in order to generate a homogeneous template. A matching algorithm is based on the Hamming Distance (HD) to find the similarity degree, as shown in [2].

Table 5 Achieved Experimental Results

Biometric System	Database	FAR	FRR
Unimodal (Fingerprints)	FVC2002 DB2A-S1	2.86%	17.64%
	FVC2002 DB2A-S2	0.23%	11.77%
Unimodal (Iris)	BATH-S2	1.19%	11.04%
	BATH-S3	1.71%	12.67%
Multimodal	DBtest2	0%	8.19%
	DBtest3	0%	9.7%
	DBtest4	0%	7.28%
	DBtest5	0%	8.16%

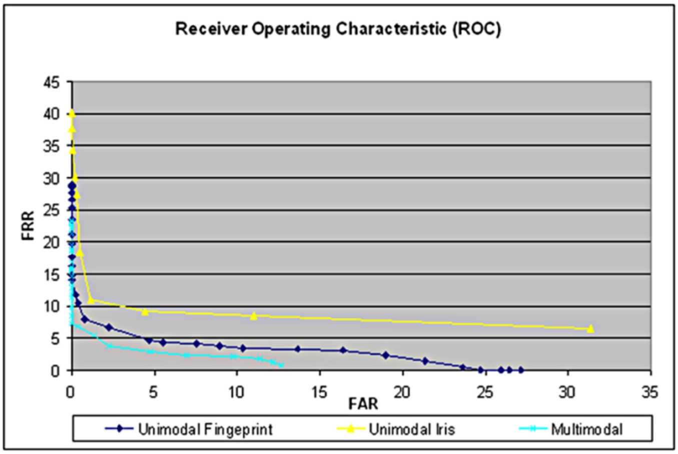


Fig. 12 ROC curve relative to DBtest4 tests

Prototype Accuracy

Table 5 shows the experimental results achieved, in terms of FAR and FRR, in unimodal and multimodal recognition systems. The table gives the possibility of being able to easily and quickly compare the results obtained with different databases, showing a good level of robustness both in terms of the approach used and obtained results.

For most completeness Figure 12 shows ROC curve relative to tests with database DBtest4. The other curves, obtained with other databases, reported similar characteristics.

Subsequently, the strategy fusion proposal has been applied and evaluated using databases of smaller size. The achieved results are listed in Table 6.

Table 6 Test Databases

Biometric System	Database	FAR	FRR
Unimodal (Iris)	BATH-S1	0.67%	9.98%
Unimodal(Fingerprint)	FVC2002_DB2B	1.35%	16.78%
Multimodal	DBtest1	0%	5.71%

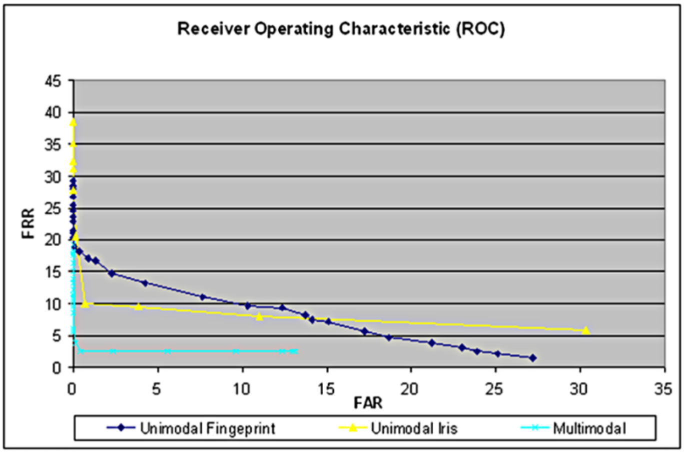


Fig. 13 ROC curve relative to other DB tests

Figure 13 shows that experimental results obtained by the multimodal recognition system are maintained below an acceptable threshold, and also comparable to the previous tests, this is a sign of a good robustness of system and approach used. This result shows a good performance considering also that for this type of fusion (the template level) there is no weight assigned to the individual results of each multimodal system.

Computational Costs

The systems have been developed using the prototyping Matlab environment on a general purpose computer with an Intel P4@3.00GHz with 2GB of RAM. Table 7 shows the average execution time for both phases of preprocessing (fingerprint and iris) and matching algorithm (which for the considered approach is the same for both systems - single mode and multimode). In the early preprocessing stage the biometric identifier extraction phase has also been considered.

Table 7 Computational Time

Stages	Fingerprints	Irises
Pre-processing	4.60 sec	3.56sec
Matching	0.37 sec	

9.4 The Embedded Multimodal Sensor for Fingerprint Recognition

An Automatic Fingerprint Authentication System (AFAS) consists of three main processing steps: image acquisition, feature extraction, and biometric template matching. In the first phase, a sensor scans and acquires the fingerprint image. Successively a

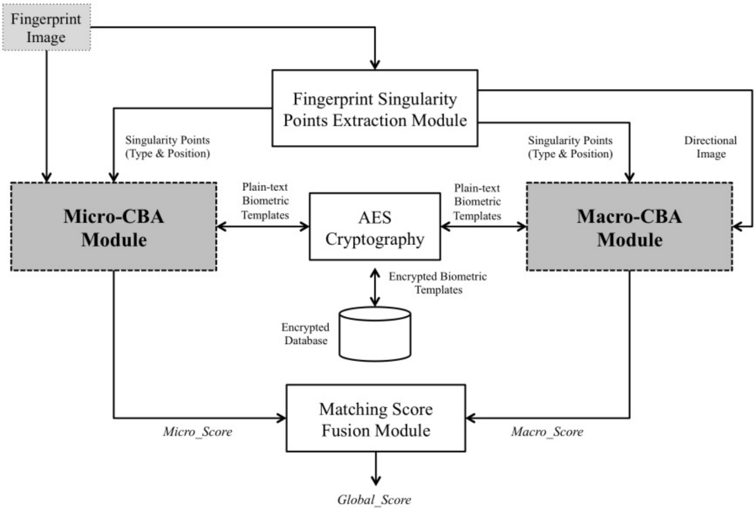

Fingerprint Image
Fingerprint Singularity Points Extraction Module
Singularity Points (Type & Position)
Singularity Points (Type & Position)
Directional Image
Micro-CBA Module
Plain-text Biometric Templates
AES Cryptography
Plain-text Biometric Templates
Macro-CBA Module
Encrypted Biometric Templates
Encrypted Database
Matching Score Fusion Module
Micro_Score
Macro_Score
Global_Score

Fig. 14 The described block scheme of proposed the multimodal authentication system. The gray blocks draw attention to the main modules of the proposed system: the Micro-Characteristics Based Authentication Module (MicroCBA Module) and the Macro-Characteristics Based Authentication Module (MacroCBA Module).

vector of features, containing information about the micro and/or the macro features will be extracted. In many cases, this step is preceded by a pre-processing phase in order to enhance fingerprint image quality. Finally, a matching score is used to quantify the similarity degree between the input image and the stored templates. Generally, a threshold based process is used to accept or reject a user.

The proposed system architecture is composed of two AFAS modules based on micro and macro features, respectively. Result fusion is realized combining the matching score of both AFASs in order to obtain an overall matching score.

As depicted in Figure 14, the Fingerprint Singularity Points Extraction Module processes an acquired fingerprint image in order to extract useful information (presence, number, and position) on *core* and *delta* points.

Fingerprint image as well as singularity point information is used as inputs of both Micro-CBA and Macro-CBA Modules. The first one uses singularity point information for fingerprint registration and performs fingerprint templates matching using minutiae type and position (micro-features), while the second one performs fingerprint templates matching using only the directional image of the original fingerprint and the singularity point information. Fingerprint templates are encrypted before their storage. The unimodal matching scores are finally combined to obtain the overall matching score. However, singularity point detections can fail, since fingerprint could be corrupted, broken or the fingerprint has not core and delta points (i.e. it belongs to the Arch class). In this case, the Micro-CBA Module performs fingerprint templates matching using only minutiae information without fingerprint registration, while the Macro-CBA Module will give zero as matching. For this reason, the overall matching score is obtained using different weights for the two AFASs.

The Minutiae-Based Matching Algorithm

An ideal matching system should be immune to fingerprint translation, rotation and non-linear deformation issues. For this reason, singularity point information is checked before running the fingerprint matching algorithm. As pointed out before, singularity points presence and position could be used for fingerprint pair registration before evaluating the matching score. However, if no singularity points are extracted, the template matching algorithm will be performed on the set of extracted minutiae without any registration step and fingerprint deformation reduction.

With more details, template matching algorithm is based on extracted micro-features (minutiae spatial coordinates and ridge direction) and involves a fingerprint pair composed by the acquired fingerprint and the stored template. So, the on-line acquired fingerprint image is tentatively registered. Successively, a window, centered on the minutiae position, is considered to reduce deformation problems, when and only when *core* and *delta* points are detected.

Finally a comparison between correspondent windows in each fingerprint pair is performed. The *Micro_Score* will be the percentage of correct matched minutiae.

The Singularity Points-Based Matching Algorithm

The proposed algorithm receives as input the coordinates of the core and delta points and the directional image. If no singularity points have been detected and extracted, the matching score will be equal to zero. Otherwise, singularity regions will be analyzed through singularity regions analysis and topological relations analysis:

- *Singularity region analysis*: the algorithm receives the directional image and the decrypted stored template as inputs. If the fingerprint-template pair has at least two singularity points of the same type (core-core or delta-delta), the procedure starts with an error computation, depending on the two directional image differences;
- *Topological relation analysis*: If the fingerprint-template pair has at least four singularity points, the procedure starts selecting singularity point neighborhoods with the minimum relative distance (the minimum distance is chosen to reduce distortion effects). The Euclidean distances between the same type of singularity points (core-core or delta-delta) are then calculated. If these distances are under a tolerance threshold (reduced distortion effects), all possible relative distances between the selected singularity points are computed and analyzed to extract the error measure (the pair with the lowest error is chosen). Since the distance between two singularity points is invariant with respect to roto-translations changes, the singularity point extraction can be performed without an images registration phase. The previous error measures are then combined to obtain the overall *Macro_Score* index. Since procedures and techniques to obtain the single error measure are different, a weighted sum of these errors is implemented, using the values depicted in Table 8, to obtain the overall *Macro_Score* index (the analysis of topological relations gives the highest contribution since it involves topological information of the core and delta regions).

Table 8 The used weights in the Macro-CBA module error measures. The measures come from the core analysis procedure, delta analysis procedure, and topological relation analysis procedure.

Process	**Core analysis**	**Delta analysis**	**Topological relation analysis**
Weight value	0.35	0.20	0.45

The Matching Score Level Fusion Module

The Matching Score Level Fusion Module computes the overall matching score combining the two unimodal subsystem matching scores. Since the Micro-CBA Module and the Macro-CBA Module are based on different techniques and parameters to determine the unimodal matching score, a weighted sum, with two different weights, has been used to obtain the overall matching score.

Experimental trials have demonstrated that the best performance, in terms of FAR and FRR indexes, is obtained using the following formula:

$$Global_{score} = 0.6 * Micro_{score} + 0.4 * Macro_{score} \tag{2}$$

Prototype Accuracy

The used FVC2002/DB2B database [33] is composed by gray scale fingerprint images captured by an optical sensor (see Table 1). This database is composed of 80 images of 296x560 pixels, collected from 10 people (8 acquisitions for each person).

Table 9 shows the FAR and FRR indexes obtained by the Micro-CBA module and the Macro-CBA module, respectively. In the same table the FAR and FRR indexes of the multimodal system are also listed.

Table 9 Recognition results of the unimodal Micro-CBA module, of the unimodal Macro-CBA module, and of the final multimodal system

Authentication rates on FVC database	**FAR (%)**	**FRR (%)**
Macro-CBA Module	2.56	18.92
Micro-CBA Module	1.52	20.35
Multimodal System	**1.07**	**10.71**

FAR and FRR of Micro-CBA Module have been obtained using 12 coincident minutiae for each processed fingerprint pair, as suggested by the FBI. FAR and FRR of Macro-CBA Module have been computed when the 60% of the directional field of each processed pair is coincident [8].

Multimodal system enhanced accuracy is due to the possibility to correct the wrong results of the first unimodal system using the results achieved by the second unimodal system and vice versa.

Hardware implementation precision shows the same accuracy of the corresponding software system in terms of FAR and FRR. However, fingerprint processing on the hardware device shows an overall 8x speed-up factor (see Table 10).

Execution Times

The algorithm implementation on FPGA achieves the performance of highly competitive systems. The proposed recognition system takes advantage of FPGA technologies and introduces interesting characteristics considering algorithms used and performance achieved. Table 10 shows the execution times necessary to perform every single authentication task with a working frequency of 25.175 MHz.

Table 10 Execution times of each phase and the relative speed-up factor. The working frequency is 25.175 MHz.

Module	**Pre-processing (ms)**	**Matching (ms)**	**Total time (ms)**	**Speed-Up Factor**
Micro-CBA	514.1	3.62	517.72	8X
Macro-CBA	34.8		38.42	3X

The speed-up factors refer to the number of cycles of a general purpose Intel P4@3.00GHz with 2 GB of RAM. The low working frequency suggests interesting considerations for the employment on the embedded recognizer in portable devices, since one of the techniques used to reduce device power consumption is to have a low working frequency with an adequate processing time for the device.

10 Comparisons and Discussions

In this chapter two multimodal biometric identification systems have been presented. With more details, a template-level fusion method for a multimodal biometric system based on fingerprints and irises has been described. The used approach for fingerprint and iris segmentation, coding, and matching has been tested using the official FVC2002 DB2A fingerprint database and the BATH iris database. Even if, the frequency-based approach, using fingerprint (pseudo) singularity point information, introduces an error on system recognition accuracy, the achieved recognition results have shown an interesting performance if compared with the literature approaches on similar datasets. On the other hand, in the frequency-based approach, it is very difficult to use the classical minutiae information, due to its great number. In this case, the frequency-based approach should consider a high number of ROIs, resulting in the whole fingerprint image coding, and consequently, in high-dimensional feature vector.

In order to test the effectiveness of the described multimodal approach, several datasets have been used. Firstly, two different multimodal systems have been tested and compared with the standard FVC2002 DB2B fingerprint image database and the BATH-S1 iris image database: the former was based on a matching-score-level fusion technique, while the latter was based on the proposed template-level fusion technique. The obtained results show that the proposed template-level fusion technique carries out an enhanced system showing interesting results in terms of FAR and FRR indexes. The aforementioned result suggests that the template-level fusion gives better performance than the matching-score-level fusion. This statement confirms the results presented in [38]. In this paper, Khalifa and Amara proposed the results of four different fusions of modalities at different levels for two unimodal biometric verification systems, based on offline signature and handwriting. However, the better result was obtained using a fusion strategy at the feature-extraction level. In conclusion, when a fusion strategy is performed at the feature-extraction level, a homogeneous template is generated, so that a unified matching algorithm is used, at which time the corresponding multimodal identification system shows better results when compared to the result achieved using other fusion strategies. Lastly, several 50-users' databases have been generated, combining the available FVC2002 DB2A fingerprint database and the BATH iris database. The achieved results show uniform performance on the used datasets. In literature, few multimodal biometric systems based on template-level fusion have been published, rendering it is very difficult to comment and analyze the experimental results reported in this chapter.

Concerning the iris identification system, the achieved performance can be considered very interesting when compared with the results of different approaches found in literature on the same dataset or similar dataset. A novel technique for iris recognition using texture and phase features is proposed in [37]. Texture features are extracted on the normalized iris strip using Haar Wavelet, while phase features are obtained using Log-Gabor Wavelet. The matching scores generated from individual modules are combined using the sum of their score technique. The system is tested on the BATH database giving an accuracy of 95.62%. The combined system at a matching-score-level fusion increased the system performance with FAR = 0.36% and FRR = 8.38%.

Besbes et al. [39] proposed a multimodal biometric system using fingerprint and iris features. They use a hybrid approach based on: 1) fingerprint minutiae extraction and 2) iris template encoding through a mathematical representation of the extracted iris region. However, no experimental results have been reported in their paper.

F. Yang et al. [40] proposed a mixed multimodal system based on features fusion and matching-score fusion. The paper presents the overall result of the entire system of self-constructed, proprietary databases. The paper reports the ROC graph with the unimodal and the multimodal system results. The ROC curves show the improvements introduced by the adopted fusion strategy. No FAR and FRR values are reported.

Table 11 Comparison between multimodal systems recognition rates between our and literature ones

System	Database	Feature	Fusion Level	FAR (%)	FRR (%)	EER (%)
Mehrotra et al. [37]	BATH	IRIS	Score Level	0.36	8.38	N/A
Khalifa et al. [38]	Proprietary DBs	Signature + Handwriting	Decision Level	N/A	N/A	3.80
		Signature + Handwriting	Score level	N/A	N/A	2.65
		Signature + Handwriting	Feature Extraction level	N/A	N/A	2.60
		Signature + Handwriting	Signal level	N/A	N/A	6.05
Our Systems	FVC2002 + BATH subset	Iris + Fingerprint	Template level	0	5.71	2.36
	FVC2002 DB2A + BATH	Iris + Fingerprint	Template level	0	7.28÷9.7	3.17÷5.76

Table 11 reports the comparison between multimodal systems recognition rates between our and literature ones.

Moreover, in this chapter, an embedded biometric sensor prototyped on FPGA has been outlined. The embedded recognizer, using FPGA technology, a smart-card read/write device, and the AES algorithm to cipher the biometric template, shows interesting results in terms of recognition rates. So, fingerprint processing modules have been implemented in a hardware embedded device to hide several biometric authentication system attack points [7], [8]. The prototyped access-point has been tested using the official FVC2002_DB2B fingerprint database and two proprietary databases, made through two different sensor technology acquisitions So, fingerprint processing modules have been implemented in a hardware embedded device to hide several biometric authentication system attack points [7], [8]. The prototyped access-point has been tested using the official FVC2002_DB2B fingerprint database and two proprietary databases, made through two different sensor technology acquisitions. For example, in [41], the authors proposed an implementation of a hardware identification system. However, the fingerprint matching phase was not developed and presented, so that no direct comparison with this work can be made. The remaining fingerprint processing tasks were implemented in an FPGA device with a clock frequency of 27.65 MHz and a processing time of 589.6 ms. Compared with this system, the achieved execution times denote a high performance level.

In [41] Bonato et al. proposed a hardware system using a pipeline technique to increase the final output. The needed fingerprint pre-processing tasks had been implemented on the Altera FLEX10KE FPGA. Initially, a Gaussian filtering is used to enhance fingerprint quality and an edge-detection algorithm is applied to segment fingerprint ridges. Finally, a thinning algorithm is applied before minutiae localization. The processing time takes 306.93 ms. However, fingerprint matching time was not reported by the authors.

In [42] Schaumont et al. developed an embedded device for fingerprint matching operations. ThumbPod uses multiple levels of programming (Java, C and hardware) with a hierarchy of programmable architectures (KVM on top of a SPARC core on top of an FPGA). The bottom system layer is composed of a Xilinx Virtex-II XC2V1000 FPGA, on which the authors configured a soft-core processor with two hardware co-processors. In addition an encryption processor and a Discrete Fourier Transform signal processor provide acceleration for the computational critical components. The authors evaluated its implementation by means of FAR (0.01%) and FRR (0.5%) indexes. However, no description and details of the used databases are reported in the paper. Average execution time is in the range 4-6 seconds.

In [43] Miyazawa et al. presented a DSP prototype implementation of a previously proposed phase-based matching algorithm for iris recognition. The authors implemented a compact version of their method on the hardware device and evaluated their approach using the CASIA database. The authors reported either the ROC (Receiver Operating Characteristic) curve or the EER (Equal Error Rate) index. Analyzing their results, the system shows the following indexes: EER=8.3%, FMR (False Match Rate)= 0%, FNMR (False Non-Match Rate)=25%.

The overall iris recognition procedure takes 1 second on the DSP device (Texas Instruments TMS320DM641, 400MHz).

In [44] the design of embedded multimodal biometric systems is analyzed by Yoo et al. The authors have implemented a real-time system using a hardware platform composed of a low power ARM920T S3C2440A (400MHz) core processor and a connected Xilinx XC3S4000 FPGA. Initially, the system was implemented on the ARM processor and then the most time-consuming biometric system components were implemented on FPGA. They used the ETRI face database, the CASIA V.1.0 iris image database, and the FVC 2004 DB3 fingerprint database to test the performance of each single unimodal biometric system by mean of the EER index. The presented results show EER=1.50% of the face-based identification system, EER=1.68% for the iris-based identification system, and EER=4.53% for the fingerprint-based identification system. Execution times are 1.2 Sec, 1.0 Sec, and 1.8 Sec, respectively. No fusion techniques and results were presented in the multimodal system.

Table 12 reports the comparison between embedded systems recognition rates between our and literature ones.

Table 12 Comparison between embedded systems recognition rates between our and literature ones

Sensor	Type	Trait	Database	FAR (%)	FRR (%)	ERR (%)
Bonato et al. [41]	Unimodal	Fingerprint	Proprietary	N/A	N/A	N/A
Schaumont et al. [42]	Unimodal	Fingerprint	Proprietary	0.01	0.50	N/A
Miyazawa et al. [43]	Unimodal	Iris	CASIA	0.00	25.00	8.30
Yoo et al. [44]	Unimodal	Face Iris Fingerprint	ETRI CASIA FVC2004	N/A	N/A	1.50 1.68 4.53
Our Sensor	Multimodal	Fingerprint	FVC2002	1.07	10.71	N/A

11 Conclusions

In the development of heterogeneous applications for workstation network, for mobile and portable systems and devices relating to all collaborative and pervasive computational technology, authentication systems represent the first step to enable data, resources, and service access protection. In this chapter, a template-level fusion algorithm working on a unified biometric descriptor has been presented. In addition, a prototyped embedded multimodal biometric sensor has been described, outlined and tested. The fingerprint/iris multimodal biometric system has been tested on different congruent datasets obtained by the official FVC2002 DB2 fingerprint database and the BATH iris database. The first test conducted on ten users has resulted in FAR = 0% and FRR = 5.71%, while tests conducted on the FVC2002 DB2A and BATH databases resulted in an FAR = 0% and an FRR = 7.28% ÷ 9.7%. The embedded biometric system has been tested on the official FVC2002 DB2B fingerprint database resulting in FAR = 1.07% and FRR = 10.71%. The final embedded multimodal biometric prototype shows a speed-up factor of 8x respect to software implementation and showing the same accuracy in terms of FAR and FRR. On the other hand, Internet of Things will involve a heterogeneous set of interacting devices to enable innovative global and local applications and services for users. The authentication 'thing' will be an embedded sensor able to acquire and compare biometric traits for user authentication. However, the use of heterogeneous devices should evolve into a more structured set of solutions, where 'things' communicate with other entities and they are closely integrated with Internet hosts, infrastructure, and services.

References

1. Karume, S.M., Omieno, K.K.: Synergizing E-infrastructures Initiatives to Foster e-Research in HigherEducation Institutions in Africa. Journal of Emerging Trends in Computing and Information Sciences 2(11), 632–640 (2011) ISSN 2079-8407
2. Conti, V., Militello, C., Sorbello, F., Vitabile, S.: A Frequency-based Approach for Features Fusion in Fingerprint and Iris Multimodal Biometric Identification Systems. IEEE Transactions on Systems, Man, and Cybernetics (SMC) Part C: Applications & Reviews, 384–395, doi:10.1109/TSMCC.2010.2045374, ISSN: 1094-6977
3. Militello, C., Conti, V., Vitabile, S., Sorbello, F.: An Embedded Iris Recognizer for Portable and Mobile Devices. Special Issue on "Frontiers in Complex, Intelligent and Software Intensive Systems" of International Journal of Computer Systems Science and Engineering 25(2), 33–45 (2010)
4. Conti, V., Militello, C., Vitabile, S., Sorbello, F.: A Multimodal Technique for an Embedded Fingerprint Recognizer in Mobile Payment Systems. International Journal on Mobile Information Systems 5(2), 105–124 (2009), doi:10.3233/MIS-2009-0076, ISSN: 1574-017X
5. Vitabile, S., Conti, V., Lentini, G., Sorbello, F.: An Intelligent Sensor for Fingerprint Recognition. In: Yang, L.T., Amamiya, M., Liu, Z., Guo, M., Rammig, F.J. (eds.) EUC 2005. LNCS, vol. 3824, pp. 27–36. Springer, Heidelberg (2005)
6. Militello, C., Conti, V., Vitabile, S., Sorbello, F.: Embedded Access Points for Trusted Data and Resources Access in HPC Systems. The Journal of Supercomputing, - Special Issue on High Performance Trusted Computing 55(1), 4–27, doi:DOI: 10.1007s11227-009-0379-1, ISSN (Print): 0920-8542, ISSN (Online): 1573-0484
7. Ambalakat, P.: Security of Biometric Authentication Systems. 21st Computer Science Seminar. SA1-T1-1. Page 2, `http://www.rh.edu/~rhb/cs_seminar_2005/SessionA1/ambalakat.pdf`
8. UK Biometrics Working Group(BWG): Biometrics Security Concerns (2003)

9. Michener, J.R., Acar, T.: Security domains: key management in large-scale systems. Software IEEE 17(5), 52–58 (2000), doi:10.1109/52.877864, ISSN: 0740-7459
10. Nielsen, R., Hamilton, B.A.: Observations from the Deployment of a Large Scale PKI. In: 4th Annual PKI R&D Workshop: Multiple Paths to Trust, April 19-21. NIST, Gaithersburg MD (2005)
11. Jain, A.: On-Line Fingerprint Verification. IEEE Transaction on Pattern Analysis and Machine Intelligence 19(4), 302–314 (1997)
12. Flom, L., Safir, A.: Iris Recognition System, United States Patent No. 4,641,349, (issued March 2, 1987) U.S. Government Printing Office, Washington DC (1987)
13. Daugman, J.G.: High Confidence Visual Recognition of Persons by a Test of Statistical Independence. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11), 1148–1161 (1993)
14. Field, D.J.: Relations between the statistics of natural images and the response profiles of cortical cells. Journal of the Optical Society of America (1987)
15. Masek, L.: Recognition of human Iris patterns for biometric identification. Master's thesis, Univ. Western Australia, Australia (2003), `http://www.csse.uwa.edu.au/-pk/studentprojects/libor/` (November 2009)
16. Thai, R.: Fingerprint Image Enhancement and Minutiae Extraction. PhD Thesis, The University of Western Australia (2003)
17. Mehtre, B.M.: Fingerprint image analysis for automatic identification. Machine Vision and Applications 6(2), 124–139 (1993)
18. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Comm. ACM. 27(3), 236–239 (1984)
19. Conti, V., Militello, C., Vitabile, S., Sorbello, F.: Introducing Pseudo-Singularity Points for Efficient Fingerprints Classification and Recognition. In: 4thInternational Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2010), February 15-8, pp. 368–375. Andrzej FryczModrzewski Cracow College, Krakow (2010), doi:doi:10.1109/CISIS.2010.134
20. Karu, K., Jain, A.K.: Fingerprint classification. Pattern Recognition 29(3), 389–404 (1996)
21. Wildes, R.P.: Iris Recognition: An Emerging Biometric Technology. In: Proc. of the IEEE 85(9), 1348–1363 (1997)
22. Wildes, R.P., Asmuth, J.C., Green, G.L., Hsu, S.C., Kolczynski, R.J., Matey, J.R., McBride, S.E.: A System for Automated Iris Recognition. In: Proc. of the 2nd IEEE Workshop Applications of Computer Vision, Sarasota, FL, December 5–7, pp. 121–128 (1994)
23. Sanchez-Reillo, R., Sanchez-Avila, C., de Martin-Roche, D.: Iris Recognition for Biometric Identification using Dyadic Wavelet Transform Zero – Crossing. In: 2001 IEEE 35th International Carnahan Conference on Security Technology, London, October 16-19, pp. 272–277 (2001)
24. Sanchez-Reillo, R., Sanchez-Avila, C.: Iris Recognition with Low Template Size. In: Proc.of International Conference Audio and Video – Based Biometric Person Authentication, pp. 324–329 (2001)
25. Sanchez-Reillo, R., Sanchez-Avila, C., de Martin-Roche, D.: Iris – Based Biometric Recognition using Dyadic Wavelet Transform. IEEE Aerospace and Electronic Systems Magazine, 3–6 (October 2002)
26. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 679–698
27. PohHoonThian, N., Bengio, S., Korczak, J.: A Multi-Sample Multi-Source Model For Biometric Authentication. In: Proc. of IDIAP (April 2002)

28. Jain, A.K., Hong, L., Kulkarni, Y.: A Multimodal Biometric System Using Fingerprint, Face and Speech. In: Conference on Audio-Video based Biometric Person Authentication (1999)
29. Bubeck, M.: MultibiometricAutentication. Term Project CS574, San Diego State University (Spring 2003)
30. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer (2003)
31. Ross, A., Jain, A.: Information fusion in biometrics. Pattern Recognition Letters 24, 2115–2125 (2003)
32. Daugman, J.: How iris recognition works. IEEE Trans. Circuits Syst. Video Technol. 14(1), 21–30 (2004), doi:10.1109/TCSVT.2003.818350
33. FVC (2002), `http://bias.csr.unibo.it/fvc2002/databases.asp`
34. BATH, `http://www.bath.ac.uk/eleceng/research/sipg/irisweb/`
35. Australian Society of Ophthalmologists, `http://www.aso.asn.au/`
36. Castro, M., Jara, A.J., Skarmeta, A.F.: An analysis of M2M platforms: challenges and opportunities for the Internet of Things. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (2012)
37. Mehrotra, H., Majhi, B., Gupta, P.: Multi-algorithmic Iris authentication system. Presented at the World Acad. Sci., Eng. Technol., Buenos Aires, Argentina 34 (2008) ISSN 2070-3740
38. Khalifa, A.B., Amara, N.E.B.: Bimodal biometric verification with different fusion levels. In: Proc. 6th Int. Multi-Conf. Syst., Signals Devices, SSD 2009, pp. 1–6 (2009), doi:10.1109/SSD.2009.4956731.
39. Besbes, F., Trichili, H., Solaiman, B.: Multimodal biometric system based on fingerprint identification and Iris recognition. In: Proc. 3rd Int. IEEE Conf. Inf. Commun. Technol.: From Theory to Applications (ICTTA 2008), pp. 1–5 (2008), doi:10.1109/ICTTA.2008.4530129
40. Yang, F., Ma, B.: A new mixed-mode biometrics information fusion based-on fingerprint, hand-geometry and palm-print. In: Proc. 4th Int. IEEE Conf. Image Graph, pp. 689–693 (2007), doi:10.1109/ICIG.2007.39
41. Bonato, L.V., Molz, R.F., Furtado, J.C., Ferro, M.F., Moraes, F.G. Design of a fingerprint system using a hardware/software environment. In: Proc. of the 2003 ACM/SIGDA 11th International Symposium on Field Programmable Gate Arrays (2003a) ISBN:1-58113-651-X
42. Schaumont, P., Sakiyama, K., Fan, Y., Hwang, D., Yang, S., Hodjat, A., Lai, B., Verbauwhede, I.: Testing ThumbPod: Softcore bugs are hard to find. In: 8th IEEE International High-Level Design Validation and Test Workshop, pp. 77–82 (2003) ISBN:0-7803-8236-6
43. Miyazawa, K., Ito, K., Aok, T., Kobayashi, K., Katsumata, A.: An Iris Recognition System Using Phase-Based Image Matching. In: IEEE International Conference on Image Processing, pp. 325–328 (2006)
44. Yoo, J.H., Ko, J.G., Chung, Y.S., Jung, S.U., Kim, K.H., Moon, K.Y., Chung, K.: Design of Embedded Multimodal Biometric Systems. In: 3rd International IEEE Conference on Signal-Image Technologies and Internet-Based System, pp. 1058–1062 (2007), doi:10.1109/SITIS.2007.130
45. Daemen, J., Rijmen, V.: AES proposal: Rijndael. From web `http://csrc.nist.gov/encryption/aes/round2/AESAlgs/Rijndael/Rijndael.pdf`

Author Index

Subject Index