Journal of Computer Networks and Communications

Context-Enriched and Location-Aware Services

Guest Editors: MoonBae Song, Goreti Marreiros, Hoon Ko, and Jae-Ho Choi

Context-Enriched and Location-Aware Services

Journal of Computer Networks and Communications

Context-Enriched and Location-Aware Services

Guest Editors: MoonBae Song, Goreti Marreiros, Hoon Ko, and Jae-Ho Choi

Editorial Board

Contents

Context-Enriched and Location-Aware Services, MoonBae Song, Goreti Marreiros, Hoon Ko, and Jae-Ho Choi
Volume 2012, Article ID 649584, 2 pages

Enabling Semantic Technology Empowered Smart Spaces, Jussi Kiljander, Arto Ylisaukko-oja, Janne Takalo-Mattila, Matti Eteläperä, and Juha-Pekka Soininen
Volume 2012, Article ID 845762, 14 pages

A Fusion Approach of RSSI and LQI for Indoor Localization System Using Adaptive Smoothers, Sharly Joana Halder and Wooju Kim
Volume 2012, Article ID 790374, 10 pages

A DHT-Based Discovery Service for the Internet of Things, Federica Paganelli and David Parlanti
Volume 2012, Article ID 107041, 11 pages

Route Anomaly Detection Using a Linear Route Representation, Wen-Chen Hu, Naima Kaabouch, Hung-Jen Yang, and S. Hossein Mousavinezhad
Volume 2012, Article ID 675605, 12 pages

Context-Aware Adaptation of Component-Based Systems: An Active Repository Approach, Sindolfo Miranda Filho, Julio Melo, Luiz Eduardo Leite, and Guido Lemos
Volume 2012, Article ID 963728, 11 pages

Exploiting Location and Contextual Information to Develop a Comprehensive Framework for Proactive Handover in Heterogeneous Environments, G. Mapp, F. Katsriku, M. Aiash, N. Chinnam, R. Lopes, E. Moreira, R. M. Porto Vanni, and M. Augusto
Volume 2012, Article ID 748163, 17 pages

Enhancing Existing Communication Services with Context Awareness, Bachir Chihani, Emmanuel Bertin, Irsalina Salsabila Suprapto, Julien Zimmermann, and Noël Crespi
Volume 2012, Article ID 493261, 10 pages

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 649584, 2 pages
doi:10.1155/2012/649584

Editorial
Context-Enriched and Location-Aware Services

MoonBae Song,[1] Goreti Marreiros,[2] Hoon Ko,[2] and Jae-Ho Choi[3]

[1] *Mobile Communications Division, Samsung Electronics Co., Ltd., 416 Maetan-dong, Youngtong-gu, Suwon, Gyeonggi-do 443-742, Republic of Korea*
[2] *Knowledge Engineering & Decision Support Research Group (GECAD), Institute of Engineering-Polytechnic of Porto (ISEP/IPP), Rua Dr. Antonio Bernardino de Almeida 431, 4200-072 Porto, Portugal*
[3] *Department of Computer Science, College of Engineering, Yonsei University, 134 Shinchon-dong, Seodaemoon-gu, Seoul 120-749, Republic of Korea*

Correspondence should be addressed to MoonBae Song, mbsong@gmail.com

Received 9 December 2012; Accepted 9 December 2012

Context-enriched and location-aware services (CELAS) effect the current customer-facing mobile applications and web services in our daily lives such as Facebook, Google, and Twitter. This special issue on Context-Enriched and Location-Aware Services compiles 6 exciting papers among many well-written manuscripts, most of which are very carefully reviewed.

The paper titled "*A fusion approach of RSSI and LQI for indoor localization system using adaptive smoothers*" reviewed by Dr. Song introduces an indoor localization technique that exploits both Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) together in order to reduce location error.

Dr. Choi reviews "*A DHT-based discovery service for the Internet of Things*" which aims to design a novel discovery service for the Internet of Things for smart space. The work envisages the conception of novel services and implemented various prototype applications.

The paper titled "*Route anomaly detection using a linear route representation*" discusses a technique for detecting route anomaly which is a common problem in our daily lives. The proposed method is based on a linear route representation that finds the matched routes from a set of stored routes as the current route is entered location by location. An alert may be generated from the detected route anomaly, which is very practical in our daily lives.

Professor Marreiros reviews "*Context-aware adaptation of component-based systems: an active repository approach*" which proposes context-aware adaptation technique for component-based software (CBS) systems. In this paper, the authors propose an active component repository that is able to receive the current configuration and components from the context-aware system and the new architecture that better fits the given context. Considering the repository with a wide knowledge of available components, more suitable configuration for the current running system can be achievable.

Dr. Ko reviews "*Exploiting location and contextual information to develop a comprehensive framework for proactive handover in heterogeneous environments.*" This paper aims to explore the development of a comprehensive framework for achieving optimal communication in heterogeneous wireless environments using location and contextual information to provide efficient handover mechanisms. By using location-based techniques, they make it possible to demonstrate that the Time Before Vertical Handover as well as the Network Dwell Time can be accurately estimated including how location and context awareness can be used to estimate the best handover radius. The paper highlights the use of ontological techniques as a mechanism for specifying and prototyping such systems.

Dr. Song reviews "*Enhancing existing communication services with context awareness*" which introduces a context aware communication system that takes into account user's preferences, workload, and situation to deliver customized telephony services. The authors provide a reference implementation with IMS platform in which context awareness features such as incoming call management are able to be provided without modifying the user experience.

By compiling these papers, along with various topics in CELAS such as indoor localization, contextualized adaptation, and service/anomaly detection, we hope to enrich our readers and researchers with respect to the most interesting field of research that is fruitful in both academic society and industrial society.

MoonBae Song
Goreti Marreiros
Hoon Ko
Jae-Ho Choi

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 845762, 14 pages
doi:10.1155/2012/845762

Research Article

Enabling Semantic Technology Empowered Smart Spaces

Jussi Kiljander, Arto Ylisaukko-oja, Janne Takalo-Mattila, Matti Eteläperä, and Juha-Pekka Soininen

VTT Technical Research Centre of Finland, Kaitoväylä 1, Oulu, P.O. Box 1100, 90571 Oulu, Finland

Correspondence should be addressed to Jussi Kiljander, jussi.kiljander@vtt.fi

Received 16 March 2012; Revised 17 September 2012; Accepted 19 September 2012

Academic Editor: Jae-Ho Choi

It has been proposed that Semantic Web technologies would be key enablers in achieving context-aware computing in our everyday environments. In our vision of semantic technology empowered smart spaces, the whole interaction model is based on the sharing of semantic data via common blackboards. This approach allows smart space applications to take full advantage of semantic technologies. Because of its novelty, there is, however, a lack of solutions and methods for developing semantic smart space applications according to this vision. In this paper, we present solutions to the most relevant challenges we have faced when developing context-aware computing in smart spaces. In particular the paper describes (1) methods for utilizing semantic technologies with resource restricted-devices, (2) a solution for identifying real world objects in semantic technology empowered smart spaces, (3) a method for users to modify the behavior of context-aware smart space applications, and (4) an approach for content sharing between autonomous smart space agents. The proposed solutions include ontologies, system models, and guidelines for building smart spaces with the M3 semantic information sharing platform. To validate and demonstrate the approaches in practice, we have implemented various prototype smart space applications and tools.

1. Introduction

The environments we live in (homes, cars, work places, etc.) are inhabited by a large and a constantly increasing number of electronic devices with huge amounts of information embedded into them. Smart space is a name for a physical place where these devices interoperate with each other in order to provide the user with services that are relevant in the given situation. In order to achieve this, it is necessary for a device to be able to "understand" its context. Here the term context follows the definition: "*Context is any information that can be used to characterize the situation of an entity.*" [1].

The smart space vision can be traced back to the beginning of the 1990s, when Mark Weiser presented his ideas of ubiquitous computing [2]. In addition to ubiquitous computing, smart spaces are a widely studied concept (with a slightly different area of emphasis) in pervasive computing [3], ambient intelligence (AmI) [4], and Internet of Things (IoT) [5] research. There have been many projects such as Buxton's Reactive Environment [6], Massachusetts Institute of Technology's Oxygen [7], Microsoft's EasyLiving [8], Hewlett Packard's Cooltown [9], and Stanford University's iRoom [10], just to name a few, focusing on different aspects of smart spaces. These research projects, among others, have produced many approaches for realizing various features of context-aware computing in smart spaces. However, to enable smart spaces on a larger scale, there is a need for Web-like infrastructure that provides a reusable base for building context-aware smart space applications.

Service-oriented architecture (SOA) frameworks and technologies such as Universal Plug and Play (UPnP) [11], OSGi [12], Simple Object Access Protocol (SOAP) [13], and Representational State Transfer (REST) [14] represent progress to the right direction as they provide reusable solutions for interoperability. However, SOA technologies are still heavily focused on using case specific a *priori* standardization that has its limits as it is impossible to anticipate all possible needs of future applications. Additionally, SOA technologies do not provide a standard language for presenting semantics of information in a formal structured manner. Because of this, it is difficult to develop SOA-based systems where devices share a mutual understanding of the context and are thus able to interoperate with each other in order to provide relevant services for the end-users.

In 2001 Berners-Lee et al. presented a landmark paper titled “The Semantic Web” [15]. In the paper they presented a vision of the next generation World Wide Web (WWW), where the semantics of information would be presented in machine-interpretable format allowing autonomous software components, called agents, to execute tasks on behalf of humans. As semantic technologies provide, at the same time, a very flexible linked data [16] model and mechanisms for interchanging semantics of information in a structured format, they seem to be also a natural fit to the interoperability needs of context-aware smart space applications. This has been also proposed in various occasions. For example, in [17] Chen presents a Context Broker Architecture for Pervasive Computing (CoBrA) that utilizes Semantic Web technologies to provide context-aware computing infrastructure for physical places, and in [18] Wang et al. describe Semantic Web-based pervasive computing infrastructure called Semantic space.

The possibilities provided by the Semantic Web technologies to context-aware smart space systems would be plenty. Ontologies presented with semantic technologies such as the Resource Description Framework (RDF) [19], the RDF Schema (RDFS) [20], and the Web Ontology Language (OWL) [21] could be used to describe the properties, capabilities, and intentions of devices, persons, and other physical objects in the environment. This would make it easier to develop context-aware applications capable of providing more relevant services for users. The flexible linked data model of RDF would make it also much easier to combine information from various sources and even from different application domains in order to create a better view of the context in the environment. More importantly, RDF would allow new information to be added without a risk of breaking the existing systems. In the most advanced scenarios, the Semantic Web technologies would even allow smart space applications to utilize information about concepts they are unfamiliar with. This is made possible by the Semantic Web ontologies that allow unknown concepts to be described in terms of known ones much in the same way as humans use encyclopedias to describe words they do not know.

We agree with the vision that context-aware smart space infrastructure should rely on ontology-based information models and utilize Semantic Web technologies in ontology presentation. However, our vision differs significantly from existing approaches such as Task Computing Environment (TCE) [22], COCOA [23], Semantic Middleware for IoT [24], and Amigo [25], where Semantic Web technologies have been used to assist the user to better exploit the services available in a pervasive computing environment. We propose that the semantic technologies are not only used to enhance service discovery, composition, and utilization, but also as a way to share any kind of information in the smart space. By this we mean that the whole interaction model in smart spaces is based on semantic information sharing via common knowledge bases. When compared with the aforementioned service-based approaches, the advantage of our approach is that it makes it possible to fully exploit the semantic technologies in ubiquitous computing systems. In our approach, we also focus on two significant aspects that have been typically neglected in semantic smart space systems: low capacity systems and object identification.

In this paper, we will describe several approaches that make it possible to realize our vision of semantic technology empowered pervasive computing systems in practice. We first identify and then present solutions to some of the most important challenges we have faced in providing context-aware computing in semantic smart space systems. The work presented in the paper is built on top of M3 semantic interoperability platform [26] (that we have been developing in various projects) and includes various approaches, ontologies, and smart space applications and tools. When combined, these approaches and methods are significant because they enable the development of smart space applications which are able to fully exploit semantic technologies. The main scientific contributions of the paper can be summarized as follows:

(1) solutions for opening information of low-capacity embedded systems for semantic technology empowered smart spaces,

(2) methods to identify real-world objects in semantic smart spaces and to fetch information related to these objects from all over the world,

(3) methods for end-user to modify behavior of semantic technology empowered smart spaces,

(4) approach for file sharing between autonomous smart space agents.

The rest of the paper is organized as follows. In Section 2, we introduce necessary background information related to the M3 concept. Section 3 presents the related work and how ours differ from it. Section 4 describes the main challenges we have faced in building semantic technology based context-aware applications to smart spaces. In Section 5, we present several solutions and approaches to overcome these challenges. Section 6 describes prototype smart spaces applications and tools we have implemented to validate our approaches in practice. Finally, Section 7 concludes the paper and presents some future work directions.

2. M3: Semantic Information Sharing Solution for Smart Spaces

The goal of M3 is to exploit the Semantic Web ideas of ontologies and linked data to provide location-aware services to physical places. In practice this is achieved through ontology-based interoperability model and a functional architecture that specifies a device, application, and service domain in independent way to access the semantic data in smart spaces.

Semantic Web standards including RDF, RDFS, and OWL provide the core technologies for the ontology-based interoperability model of M3. RDF is a Semantic Web standard for modeling metadata in form of subject, predicate, and object triples. For smart space agents, RDF triples provide a natural way to create linked data structures about the context information of smart spaces. RDFS and OWL in turn provide vocabularies to be used on top of RDF to describe relevant concepts in smart spaces as machine-interpretable ontologies. Excluding the RDFS and OWL ontologies, the M3 concept does not require any specific ontology to be used when M3 applications are developed. This allows existing Semantic Web and pervasive computing ontologies such as SSN [27], SOUPA [28], and ULCO [18] to be utilized when creating new M3 applications.

The M3 functional architecture differs from other similar frameworks such as CoBrA and Semantic Space in its simplicity as it defines only two types of processes: Knowledge Processor (KP) and Semantic Information Broker (SIB). SIB is a shared blackboard of semantic information that provides KPs with an interface for sharing semantic data. KPs are software agents whose role is to provide the end-user with services by interacting with each other via the SIB. The KPs sharing information via a common SIB form a single smart space as illustrated in Figure 1. The Smart Space Access Protocol (SSAP) defines the rules and syntax of the communication between the SIB and KP. The SSAP is based on publish/subscribe paradigm providing event based interaction via the following operations: *join()*, *leave()*, *insert()*, *remove()*, *update()*, *query()*, *subscribe()*, and *unsubscribe()*.

The basic principle of M3 is to build on top of the existing communication and service level solutions. In practice this means that SSAP can be used as a payload in any communication technology and a KP can interact with a SIB if they share at least a single common communication method. A SIB can be also implemented as a service in any SOA solution allowing SOA-specific methods to be used by KPs to discover and interact with a SIB.

There are two implementations of the M3 concept available: Smart-M3 [29] and RDF Information Base Solutions (RIBS) [30]. Smart-M3 is Linux-based implementation that utilizes XML serialized SSAP format and supports both NoTA [31] and plain TCP/IP-based communication technologies. Additionally, the Smart-M3 SIB has also been implemented as OSGi framework service [32]. In the core of the Smart-M3 SIB is a RDF++ database called *PIGLET*. The *PIGLET* provides an interface for two types of query languages: simple template queries and Wilburs query language (WQL) [33]. The template query pattern consists of a one or more triples which are matched separately against an RDF graph in a SIB. The triples may contain a wildcard *sib:any* denoting any node in the RDF graph. In WQL a query is specified in terms of a start node and a path consisting of predicates to be traversed from the start node. In this paper, WQL path queries are presented using the following syntax:

WQL(startNode, path)

RIBS is a SIB implementation designed for security, portability, and performance. It is implemented in ANSI-C programming language and it supports both plain TCP and NoTA-based transports. The security in RIBS is based on transport layer security (TLS) and triple level access control mechanisms. To provide better performance and portability, RIBS utilizes a more compact SSAP serialization format, called Word Aligned XML (WAX). RIBS also differs from the Smart-M3 in the query languages it supports. In addition to the basic template queries, RIBS provides limited SPARQL [34] support. WQL is not supported.

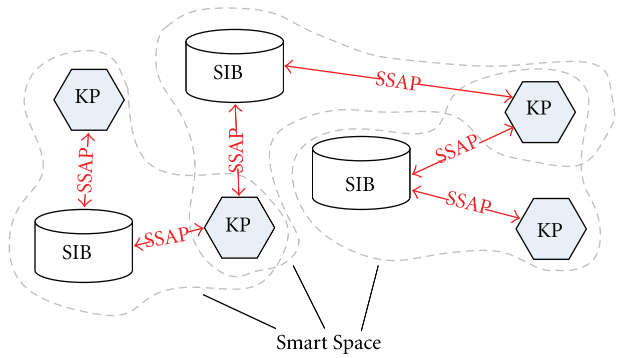


Figure 1: Composition of M3-based smart spaces.

3. Related Work

After the emergence of the Semantic Web paradigm, many semantic technology-based pervasive computing middleware systems have been proposed. The most typical way to utilize semantic technologies in pervasive computing systems is to use them to enhance traditional SOA-based systems. These approaches include, for example, TCE, COCOA, Semantic Middleware for IoT, and Amigo.

To the authors' knowledge, TCE is the first approach to combine Semantic Web and SOA technologies in pervasive computing domain. In TCE the term *Task Computing* refers to a computation that provides the end-user with required functionality by utilizing the services available in the given situation. The main goal of TCE is to allow users to concentrate on tasks the user wants to do, rather than the specific ways for accomplishing the task. A TCE system consists of one or more instances of the following components: Task Computing Client, Semantically Described Service, Semantic Service Discovery Mechanism, and Service Control (optional).

COCOA is a conversation-based solution for on-the-fly service integration in pervasive computing environments. COCOA is built on the top of OWL-S [35] and consists of three parts: COCOA-L, COCOA-SD, and COCOA-CI. The COCOA-L is OWL-S extension that provides means to specify the service capabilities, service conversation, and quality of service (QoS) properties. Service discovery and selection in COCOA are implemented in COCOA-SD. Both of the nonfunctional and functional capabilities are utilized in the discovery/selection process. The final composition of the required service is executed by COCOA-CI. The distinctive feature of COCOA-CI is that both the tasks and the integrated services are modeled as conversations.

The basic idea in the Semantic Middleware for IoT is to transform the existing specifications of standards such as Bluetooth and UPnP into Semantic Web services described with OWL-S. This enables users to create task by combining the capabilities of different devices without the need to know how the functionality is achieved in practice. There are three steps defined in the approach: device semanticization, task building, and device grounding. In the device semanticization phase, the middleware framework utilizes the standard specific discovery mechanism for locating the devices and services. Then the middleware extracts the necessary data and creates the OWL-S description at a run-time. In the task building phase, the Task Computing framework assists the user in the creation of new tasks. The last phase is the device grounding where the actual task is executed. In this phase, the technology-specific service realization is invoked to execute the required functionality of the services.

The Ambient Intelligence for the Networked Home Environment (Amigo) Project has developed an interoperability platform that enables interaction between heterogeneous services and devices in a home network domain. The Amigo architecture is based on the SOA paradigm and supports various different SOA technologies such as UPnP, Service Location Protocol (SLP), and Web Service Description Language (WSDL). Additionally, it is possible to enrich any Amigo service with Amigo-S semantic service description language which continues the work of COCOA. The advantage of utilizing Amigo-S is that it provides much more flexible and effective service discovery based-on the context than can be achieved with traditional SOA-based service discovery mechanisms. The Amigo-S is based on OWL-S and extends it with functionality needed for describing context and nonfunctional properties such as the quality of the service. In addition, the OWL-S is extended with mappings to new types of groundings than just the WSDL supported by the OWL-S.

Our vision of semantic smart spaces differs significantly from the aforementioned solutions. In our approach, the semantic technologies are not only used to enhance service discovery, selection, and composition, but also as a way to share any kind of information in the smart space (i.e., the whole interaction model is based on sharing of semantic data via common knowledge bases). This kind of approach is closer to the original vision of the Semantic Web and provides better interoperability as it is not required to directly interact with various technology specific services. To concretize, the sensor information can be accessed and actuators controlled just by modifying the semantic data in the smart space. Our work also differs from the aforementioned systems by taking into account that the typical devices in smart spaces are resource restricted. Additionally, in contrast to other semantic pervasive middleware systems, we use ucode-based URIs to identify real-world objects. The ucode-based approach is more suitable to various pervasive computing scenarios such as retail, than the traditional URL-type URIs used typically in the Semantic Web-based systems.

4. Challenges of Context-Aware Computing in Semantic Technology Empowered Smart Spaces

4.1. Challenge 1: Accessing Data of Embedded Systems. There is a huge amount of data embedded into various devices that inhabit our everyday living environments. It would be beneficial to open this information for other devices and applications in a common machine-interpretable format. This would enable devices, agents and applications to access much larger pool of data and thus obtain a better view of the context in the smart space. Consequently, smart space agents, could provide more relevant and sophisticated services for the user.

Because of the heterogeneity of devices and communication methods, it is not easy to utilize semantic technologies in real-life smart spaces. Especially the most resource-limited devices and networks are a big challenge. This is because the technologies of the Semantic Web have not been designed for low-capacity embedded systems. For example, the standard serialization formats for RDF and OWL are based on Extensible Markup Language (XML) which does not suit resource-restricted communication channels well. XML is also difficult to parse on low-capacity embedded systems. Additionally, typical ontologies are difficult to process and store in resource-constrained systems. This is because ontology designers use human readable vocabulary which is sparse and causes thus overhead to the system.

4.2. Challenge 2: Identification of Real-World Objects. In order to create context-aware applications to physical environments, it is necessary to be able to identify objects such as devices, sensors, actuators, locations, and users that form the context of the physical place. In practice this means that each object that contributes to the context of the environment must have a unique virtual identifier that can be used to refer to the object in the virtual world, that is, in the semantic database.

In typical Semantic Web-based systems, Uniform Resource Identifiers (URIs) [36] and later Internationalized Resource Identifiers (IRIs) [37] have been used as identifiers for various kinds of Web resources. This approach has been adopted from the traditional Web where the URL-type URIs have been successfully used for decades. The power of the URLs is that the Web architecture can be used to locate information related to URLs from all over the world. However; this approach does not always work in Semantic Web. This is because the agents that provide data to the Semantic Web do not typically interact with other agents directly but publish the data to semantic databases. The problem here is that because these semantic databases have a different hostname than the agent who created the resource, the Web architecture cannot be used to locate the information related to a URI. Another problem related to the traditional URL in both Semantic Web and smart spaces is that the Web architecture does not allow the same URL to point to various locations. In certain application domains such as retail and logistics, it is typical that information related to a physical object is scattered to various semantic databases and it would be, therefore, required to be able to link the object identifier to multiple addresses.

Smart spaces also set their own challenges for using URIs as identifiers for physical objects. First, URIs are typically long strings that are difficult to process in resource-restricted devices. Second, resource-restricted devices in smart spaces are not necessarily connected to the Web and thus do not have unique global address that could be used as a base when creating new URIs for resources they publish to the smart space.

4.3. Challenge 3: Taking Human Needs into Account. The grand vision of smart spaces is to make the life of people better by providing useful services when necessary. The idea is to hide sensors, actuators, and other computing entities from the user so that she/he does not have to be bothered with unnecessary technical tasks. In order to achieve this, devices and context-aware applications need to be able to know what kind of services and functionality the user requires in different situations.

User profiles are the typical way to model the desires of the end-user in semantic technology-based smart space applications. User profiles provide a subtle and feasible solution to some use of cases of context-aware computing. However, it is difficult to design profiles that are, at the same time, generic enough and still suitable for all kinds of possible situations that may take place in smart spaces. In addition to user profiles, behavioral models and machine learning have been exploited in the field of ambient intelligence [4] to provide advanced methods for recognizing user needs in different situations. However, to fully realize the visions of smart spaces, humans need also to be able to be take more active role in smart spaces. Users need to able to express what kind of functionality is required from the smart space in a given context so that the services provided to the user are in fact useful for her/him.

4.4. Challenge 4: Autonomous Content Sharing. Typical smart spaces contain devices such as smart cameras that take pictures and record video of their environments. In modern computer systems, this kind of content is typically streamed as a live stream or stored as a file. Therefore, it is essential to have common methods to enable devices and agents to share content such as files and live media streams autonomously in context-aware smart spaces. By autonomously we mean that the devices interact with each other seamlessly so that the heterogeneity of devices, communication technologies, and file sharing methods are hidden from the end-user.

This is not a simple task in semantic technology empowered smart spaces, however. The main reason for this is that the semantic technology-based information sharing infrastructures such as the M3 are not feasible for storing "nonsemantic data" such as files and cannot be used to stream live media between agents. To perform this kind of functionality, the agents need to rely on existing protocols designed for this purpose. However, there is a big variety in the file sharing methods and communication technologies used to transmit data between devices. Because of this, it is difficult both to find all available file transfer services and to interact with them to execute the required functionality. In practice this means that a device which wants to ensure that a file it is sharing is accessible to as a wide range of devices and applications as possible must advertise the file with each communication technology and file sharing protocol it supports. This, of course, causes a lot of overhead when a device supports many different file sharing and communication methods.

5. Solutions for Semantic Technology Empowered Context Awareness in Smart Spaces

5.1. Accessing Information of Embedded Systems. One can think of two fundamental approaches to overcome the challenge related to opening the information of low-capacity embedded systems for context-aware applications in semantic smart spaces. The first approach is to build gateways that transform the raw data produced by various embedded systems to a semantic format. The advantage of this approach is that it is easy to implement on the embedded system side as there is no need to support semantic technologies which always introduce some amount of overhead to the communication. The drawback is that this approach is not very scalable or flexible as a new gateway needs to be implemented or existing gateways modified for each new type of device (or more precisely new type of data).

The second approach is to implement the software agent into the low-capacity embedded system. This approach is, of course, more flexible as all devices would utilize the same semantic protocol and no gateways would be needed. Another advantage of this approach is that the embedded system could utilize the semantic data produced by other devices and thus possibly improve its functionality in some way. The drawback is, however, that there are a lot of challenges in implementing a semantic technology empowered smart space agent into a resource-restricted embedded system.

There is no single solution for developing a semantic technology-based application for resource-restricted devices, but we attempt to present the most relevant issues a developer should consider when given such a task. In M3-based systems, the SSAP has a big impact in how easy it is to implement a KP in low-capacity device and therefore we propose the WAX encoded SSAP format to be used with low capacity devices. In WAX each XML tag is 32-bit long and the element contents are aligned to the word length of the CPU. This way the WAX-encoding is more compact and much faster to parse than the original SSAP/XML encoding. Another important advantage of WAX is that it does not require an XML parser and is therefore more portable to resource-restricted devices.

One of the most important issues when designing a KP to a resource-restricted platform is predictability. By this we mean that it is important to be able to define the needed computing resources such as memory at compile time.

Because of this, it is much simpler to implement a producer KP that just publishes information to the SIB. Fortunately, the majority of resource-restricted devices are sensors which simply publish information to the SIB. If a KP is required to fetch information from the SIB, the query and subscribe operations should be executed so that they do not return too large results, which may cause buffer overflows and are slow to process. However, this is not easy because the current M3 implementations do not provide feasible mechanisms for restricting the response size of the SSAP messages. With RIBS it is possible to utilize the LIMIT sequence modifier of SPARQL to restrict the number of results returned by the SIB, but this does not help when the results are of arbitrary length.

Usually the payload of a packet is the biggest factor in the memory requirements and performance of a networked system. In semantic technology-based smart space application, the payload shared between devices is defined by ontology. Ontologies targeted for context-aware pervasive computing applications such as SOUPA [28], COBRA-ONT [38], ULCO [18], and SeMaPs [39], have unfortunately adopted the Semantic Web style to use human-readable IRIs as identifiers for the concepts defined in ontology. This, of course, makes the ontology easier to design and understand by humans but is, in the end, completely unnecessary as ontologies are meant to be languages for devices. The human-readable ontologies introduce a lot of unnecessary overhead that could be avoided if the long human-readable IRIs would be replaced, for example, by unique 64-bit integer values. We propose an approach where a dual presentation for ontologies is used. A more compact binary format ontology can be created from the traditional human-readable ontology by replacing each ontology name URI and each concept URI in the ontology by unique 32-bit integers. This way we can create 64-bit identifiers where the first 32 bits are reserved for the ontology namespace and the last 32 bits for various concepts defined in a single ontology.

5.2. Object Identification. Our approach to identify real-world objects in semantic smart spaces is based on the uID technology [40] which provides mechanisms for both unique identification of real-world objects and for retrieving information related to these objects from all over the world. In uID 128-bit expandable codes, called ucodes, are used to identify real-world objects uniquely. The idea is that ucodes can be used with any kind of tags such as Radio Frequency Identification (RFID), Near Field Communication (NFC) and Quick Response (QR) Codes. The uID address sharing architecture defines how the information related to a ucode-tagged object can be accessed. The uID architecture consists of resolution and information servers. The three-tier resolution server architecture provides mechanisms for resolving the address of application domain-specific information server that hosts information related to a ucode-tagged object.

We propose an approach where real-world objects in semantic technology empowered smart spaces are tagged with ucodes and M3 SIBs are used as uID information servers which host semantic data about the ucode-tagged objects. The key idea in the approach is to use the ucode as the URI for the virtualized physical object. This way we can create a natural link between the physical object and its virtual counterpart in semantic databases. We utilize and a propose URI scheme with a prefix "ucode" to be used to identify the ucode-type URIs.

Our approach provides various interesting possibilities for context-aware smart space applications. First, it provides new kinds of ways for people to interact with the agents of semantic smart spaces. The user is able to read sensor measurements and actuate objects, for example, by simple touching them with a tag-reader equipped smart phone. Second, it allows context-aware smart space applications to offer better functionality by collecting and utilizing information related to a physical object from various sources located even in the other side of the world. A trivial example of this occurs when a repair assistant agent provides guidance for maintenance personnel by fetching necessary information related to a broken product from the SIBs of manufacturer, user, and other maintenance companies.

5.3. Enabling Users to Modify the Behavior of Smart Spaces. To provide a solution to the challenge related to taking human needs into account, we propose a rule-based approach that enables users to define how a smart space should behave in different situations. The approach allows users to define rules that consist of input condition and actions. The input condition is defined by combining selected events with Boolean operators. The idea is that all events and actions have a dual presentation. Human readable description is used for end-users and for KPs the events and actions are expressed with RDF query and update language patterns. This way even complex events can be expressed and computed using the built-in subscribe support of the M3 platform.

The approach is divided into ontology and system models. The event and action ontology (EA-ONT) describes the data shared by the components of the system model. The main classes of the EA-ONT are *Event* and *Action*. These classes have a human-readable name and a format presented either as basic triple, WQL, or SPARQL pattern. The ranges of possible values for events and actions are presented with by *NumericRange* and *EnumeratedRange* classes. The formal ontology model is presented with the Turtle notation [41] as follows:

```
#Ontology namespaces
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix event: <http://example.org/event/>.

#Class definitions
event:Event rdf:type rdfs:Class.
event:Action rdf:type rdfs:Class.
event:Format rdf:type rdfs:Class.
event:Range rdf:type rdfs:Class.
event:TripleFormat rdfs:subClassOf event:Format.
event:WqlFormat rdfs:subClassOf event:Format.
event:SparqlFormat rdfs:subClassOf event:Format.
```

```
event:EnumeratedRange rdfs:subClassOf event:Range.
event:NumericRange rdfs:subClassOf event:Range.

#Property definitions
event:hasFormat rdfs:domain event:Event,
    event:Action.
event:hasFormat rdfs:range event:Format.
event:hasRange rdfs:domain event:Format;
    rdfs:range event: Range.
event:hasSubject rdfs:domain event:TripleFormat;
    rdfs:range rdf:subject.
event:hasPredicate rdfs:domain event:TripleFormat;
    rdfs:range rdf:predicate.
event:startNode rdfs:domain event:WqlFormat;
    rdfs:range rdfs:Resource.
event:path rdfs:domain event:WqlFormat;
    rdfs:range xsd:string.
event:spaqrlQuery rdfs:domain event:SparqlFormat;
    rdfs:range xsd:string.
event:hasElement rdfs:domain event:EnumeratedRange;
    rdfs:range xsd:string.
event:minValue rdfs:domain event:NumericRange;
    rdfs:range xsd:double.
event:maxValue rdfs:domain event:NumericRange;
    rdfs:range xsd:double.
event:step rdfs:domain event:NumericRange;
    rdfs:range xsd:double.
```

The system model, illustrated in Figure 2, defines the functional entities that provide the end-user with a possibility to modify the behavior of semantic smart spaces. The system model consists of three entities: Event Provider, Rule Handler, and Event Browser. In the M3 level, these entities are implemented as KPs that utilize the SIB for sharing semantic data with each other.

Event Providers create events to smart spaces by producing semantic information as specified by the EA-ONT. Ideally Event Providers are located in devices that provide certain information to the smart space. For example, an Event Provider can be an embedded system that measures the temperature of the room and publishes this information to the smart space. It is also possible to use a generic KP for creating events and actions based on data published by other KPs. This way it is possible to (1) create events from information published by KPs whose developers were not familiar with our approach and (2) create combined events that model data produced by various KPs as a single event or action.

A central component of the system model is the Event Browser that provides the user with a possibility to define new rules to the smart space. The Event Browser is subscribed to the available events and actions and it is therefore capable of presenting the current view of the smart space functionality for the user. When the user creates a new rule, the Event Browser passes the required data about the rule to the Rule Handler that is responsible for processing the rules.

Once the Rule Handler receives a new rule, it subscribes to the events that define the input condition for the rule. This way the Rule Handler will be notified by the SIB when data related to these events is modified. When the SIB indicates that information related to the input events has been modified, the Rule Handler evaluates whether the input condition is satisfied. If the input condition is true, the Rule Handler executes actions by updating information to the SIB as specified in the triple pattern of the actions. It is noteworthy that because the events and actions are presented as query and update language patterns, the Rule Handler does not have to be familiar with the domain-specific ontology used by the KPs.

5.4. Autonomous Content Sharing in Smart Spaces. To provide a solution to the challenge related to providing autonomous content sharing in semantic smart spaces, we model necessary information about files and file sharing methods as a common ontology and use the SIB as a semantic communication channel where this data is shared between KPs. This enables software agents first to discover the available files in the smart space and then to negotiate about the most suitable methods for transferring the file between agents. The File Sharing Ontology (FS-ONT) modeled with the RDFS vocabulary can be presented with the Turtle notation as follows:

```
@prefix fso: <http://example.org/fileSharing/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

# Class definitions
fso:File rdf:type rdfs:Class.
fso:FileRequest rdf:type rdfs:Class.
fso:SharingMethod rdf:type rdfs:Class.
fso:TcpIp rdfs:subClassOf fso: SharingMethod.
fso:Bluetooth rdfs:subClassOf fso: SharingMethod.

#Property definitions
fso:filename rdfs:domain fso:File;
    rdfs:range xsd:string.
fso:format rdfs:domain fso:File;
    rdfs:range xsd:string.
fso:modifiedDate rdfs:domain fso:File;
    rdfs:range xsd:dateTime.
fso:size rdfs:domain fso:File;
    rdfs:range xsd:integer.
fso:targetFile rdfs:domain fso:FileRequest;
    rdfs:range fso:File.
fso:hasSharingMethod rdfs:domain fso:FileRequest;
    rdfs:range fso:SharingMethod.
fso:ip rdfs:domain fso:TcpIp;
    rdfs:range xsd:string.
fso:port rdfs:domain fso:TcpIp;
    rdfs:range xsd:integer.
fso:macAddr rdfs:domain fso:Bluetooth;
    rdfs:range xsd:integer.
fso:channel rdfs:domain fso:Bluetooth;
    rdfs:range xsd:integer.
```

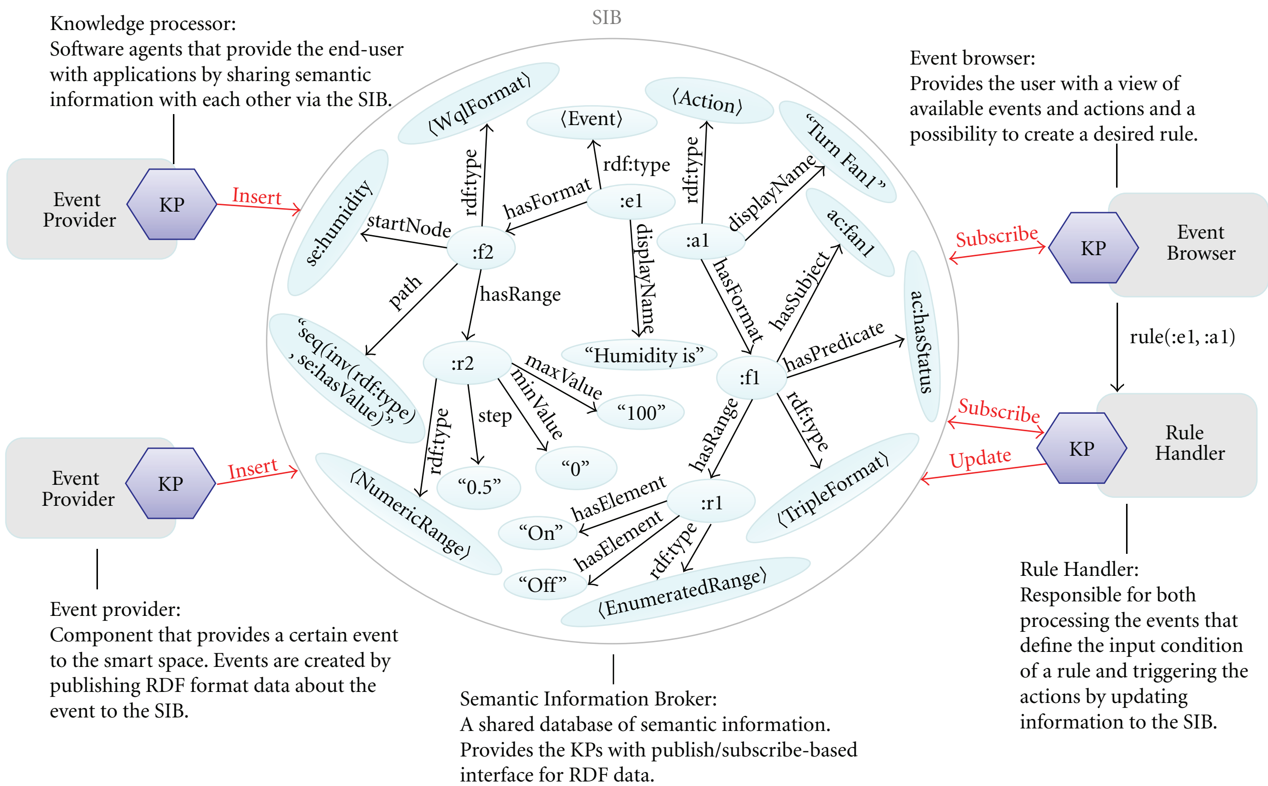


Figure 2: A system model for modifying the behavior of smart spaces.

The FS-ONT contains three main classes: *File*, *FileRequest*, and *SharingMethod*. The *File* class provides properties for describing concepts common for all kinds of files. These properties include the name, size, format, and modification date of the file. The idea is that applications that require more specific types of files import the FS-ONT and define new subclasses for the common *File* class. The *FileRequest* class represents an abstract concept of a request to a file. The class provides two properties: *targetFile* and *hasSharingMethod*. These properties associate the request with the requested file and the file sharing methods supported by the requester. The *SharingMethod* class is a base class for all protocols enabling file sharing. The FS-ONT defines two subclasses, namely, *Bluetooth* and *TcpIp*, for the *SharingMethod* class, and the idea is again that applications import the FS-ONT and introduce new subclasses when necessary.

In order to be informed about the available files and then to successfully negotiate about the most appropriate file sharing methods, KPs need to be able to perform four different SSAP operations. First, a KP that wants to share a file to a smart space needs to publish metadata about the file into the SIB as specified by the FS-ONT. Second, to be aware of the file requests, it needs to perform the following WQL subscription:

WQL(file, seq(inv(fso:targetFile),fso:hasSharingMethod))

where *file* is the URI of the *File* class instance and *seq()* and *inv()* are WQL-specific operations. The operation *seq()* denotes that the path consists of several RDF predicates and the *inv()* operation requires the predicate inside the brackets to be traversed in a reverse direction, that is, from object to subject. This operation subscribes directly to the file transfer technologies supported by KPs that request the file and thus enables the publisher KP to interpret whether the KPs share common file sharing methods.

Third, a KP that needs a certain file from the smart space needs to subscribe to all the available files in the SIB. A WQL subscription which informs when a new file is inserted to the SIB can be presented as

WQL(fso:File, inv(rdf:type)).

If needed, it is possible to make this subscription more specific when extra information such as the name or format of the file of interest is known by the requester.

Fourth, when a KP wants to request a file from another KP in the smart space, it must make a request by publishing information to the SIB as specified in the FS-ONT. As mentioned, this information must specify both the file of interest and the supported file transfer technologies. A practical example of autonomous file sharing between autonomous agents is given in Section 6.4.

6. Validation of Proposed Solutions

6.1. Smart Home Garden. The Smart Home Garden is a simple smart space where autonomous resource-restricted devices monitor the well-being of plants. It demonstrates how especially the challenges 1 and 2 can be solved in real-life smart spaces. The demonstration setup consists of potted plants, low-capacity moisture sensors, and a smart phone.

We use RIBS as the M3 broker implementation and it runs on a Via Artigo A1100 Ubuntu PC. To provide a wide range of methods for KPs to share information via the SIB, we connected a Redwire LLC *Econotag* board to the Via Artigo platform. This way the KPs can communicate with the SIB using 3G, WLAN, and IEEE 802.15.4 radio technologies.

In M3-based pervasive computing applications, ontologies play a key role in the interoperability as they define the semantics of the data shared between devices. The ontology suite developed for the Smart Home Garden consists of sub-ontologies for concepts such as sensor, plant, location, and user and can be presented with the Turtle notation as follows:

```
#Ontology namespaces
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix se: <http://example.org/sensor/>.
@prefix lo: <http://example.org/location/>.
@prefix pl: <http://example.org/plant/>.
@prefix us: <http://example.org/user/>.

# Class definitions
lo:PhysicalObject rdf:type rdfs:Class.
lo:Location rdfs:subClassOf lo:PhysicalObject.
se:Sensor rdfs:subClassOf lo:PhysicalObject.
se:Measurement rdf:type rdfs:Class.
se:MoistureSensor rdfs:subClassOf se:Sensor.
se:HumiditySensor rdfs:subClassOf se:Sensor.
se:TemperatureSensor rdfs:subClassOf se:Sensor.
se:LightSensor rdfs:subClassOf se:Sensor.
pl:Plant rdfs:subClassOf lo:PhysicalObject.
pl:PottedPlant rdfs:subClassOf pl:Plant,
    lo:Location.
us:User rdfs:subClassOf lo:PhysicalObject.

#Property definitions
se:hasMeasurement rdfs:domain se:Sensor;
    rdfs:range se:Measurement.
se:hasName rdfs:domain se:Sensor;
    rdfs:range xsd:string.
se:hasValue rdfs:domain se:Measurement;
    rdfs:range xsd:double.
se:unitOfMeasurement rdfs:domain se:Measurement;
    rdfs:range xsd:string.
lo:hasName rdfs:domain lo:Location;
    rdfs:range xsd:string.
lo:hasLocation rdfs:domain lo:PhysicalObject;
    rdfs:range lo:Location.
pl:minMoisture rdfs:domain pl:Plant;
    rdfs:range xsd:double.
pl:maxMoisture rdfs:domain pl:Plant;
    rdfs:range xsd:double.
```

A low-capacity battery powered device called the Active Tag plays a central role in the Smart Home Garden application [42]. The Active Tag is an embedded system which measures the moisture of the soil and publishes this information to the local SIB. The main task of the Active Tag is to inform the user when the soil is too dry for the given plant by blinking an LED. Each Active Tag is identified by a unique ucode located in a NFC tag connected to the device (see Figure 3). We implemented the Active Tag on a Freescale MC13224V-based *Econotag* platform. The platform runs a Contiki operating system and provides a simple IEEE 802.15.4 compatible Rime communication protocol.

The total code side of the implemented Active Tag agent including the KP, Contiki, and Rime protocol was 39.7 kB. The plain KP code including the KP application programming interface and application logic resulted in a code size of 10.3 kB. The total RAM requirement was 25.3 kB of which 14.3 kB was consumed by the Contiki operating system and the Rime protocol.

The Smart Home Garden is best explained with the following scenario. Alicia is buying new plants to her home. She does not actually have a green thumb, however, and she wants to buy plants that can be identified by the new Smart Home Garden application she acquired the day before. Alicia decides to buy two potted plants which are equipped with QR-code-based ucodes. In the uID resolution server, the ucodes of these plants point to a greenhouse SIB which contains information of the given plants.

When Alicia arrives home, the Home Garden KP in her Google Nexus S Android smart phone informs the smart space about her presence. Next, Alicia assigns the Active Tags to the plants as illustrated in Figure 3. To fetch necessary information related to the plants, Alicia points the camera of her smart phone towards the QR code in the potted plant. Once the ucode is read the Home Garden agent contacts the uID resolution server to obtain the address of the SIB which contains information about the ucode-tagged plant. After obtaining the address of the greenhouse SIB, the Home Garden KP fetches necessary information about the plants and copies the information to the local SIB in Via Artigo. Next, Alicia pairs each Active Tag with a potted plant. To do this, she touches the NFC tag in the Active Tag with her smart phone. After obtaining the ucode of the Active Tag, the Home Garden KP publishes the following RDF triple to the SIB:

```
@prefix lo: <http://example.org/location/>.
@prefix tag: <ucode:5554:2134:6442:>.
@prefix plant: <ucode:5254:2135:6325:>.
tag:1035 lo:hasLocation plant:2552.
```

This RDF triple specifies that the Active Tag identified by the ucode “5554:2134:6442:1035” is located in a potted plant identified by the ucode “5554:2134:6442:2552.” The Active Tag needs this information to obtain the moisture preference of the plant. In addition, the Active Tag KP needs information about Alicia’s presence so that it does not consume power unnecessary by blinking the LED when there is nobody to notify it. To know whether the Active Tag should blink the LED, it executes the following SPARQL ASK query to the SIB which returns a true or false response:

Figure 3: Active tag in a potted plant.

```
PREFIX lo: <http://example.org/location/>.
PREFIX pl: <http://example.org/plant/>.
ASK
{
  <5554:2134:6442:1035> lo:hasLocation ?pottedPlant.
  ?pottedPlant pl:moisturePreference ?preference.
  FILTER (?preference < value)
},
```

where *value* is the moisture value measured by the Active Tag. In addition to the query operation, the Active Tag updates the moisture value to the smart space in each cycle before going to a deep sleep mode. With 60-second wake-up intervals for query and update operations, we measured an average current consumption of 241 μA. If we do not take a battery self-discharge and LED blinking into account, this means approximate battery duration of 1.3 years with two 1.5 V and 2700 mAh alkaline batteries.

6.2. Smart Greenhouse. Smart Greenhouse is miniature version of a greenhouse where interacting agents assist a gardener in his/her daily routines. The Smart Greenhouse ecosystem illustrates how semantic technologies can be utilized to provide context awareness for more complex smart spaces than the Smart Home Garden [43, 44]. The Smart Greenhouse demonstrator, illustrated in Figure 4, consists of a miniature greenhouse, actuators, tagged plants, sensors, an autocontrol device, and a ubiquitous communicator as the gardener's personal device.

In the Smart Greenhouse, the ontologies used in Smart Home Garden are extended with new subclasses and properties for the plant class. Additionally, actuator ontology (ACT-ONT) was developed for presenting information about the physical actuators in semantic format. The ACT-ONT can be presented with the Turtle notation as follows:

Figure 4: Smart Greenhouse.

```
#Ontology namespaces
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ac: <http://example.org/actuator/>.

#Class definitions
ac:Actuator rdf:type rdfs:Class.
ac:Led rdfs:subClassOf ac:Actuator.
ac:Fan rdfs:subClassOf ac:Actuator.
ac:WaterPump rdfs:subClassOf ac:Actuator.
ac:NumericRange rdfs:subClassOf ac:Range.
ac:EnumeratedRange rdfs:subClassOf ac:Range.

#Property definitions
ac:status rdfs:domain ac:Actuator;
    rdfs:range ac:NumericRange.
ac:status rdfs:domain ac:Actuator;
    rdfs:range ac:EnumeratedRange.
```

In Smart Greenhouse, the interaction between the user and the smart space is provided by a device called the ubiquitous communicator. The ubiquitous communicator is equipped with an RFID reader and contains a Gardener Interface KP which subscribes to the sensor, plant, and actuator information to display this data for the gardener. It also enables the gardener to modify the state of actuators and provides methods for publishing information about the plants into the SIB.

The miniature greenhouse has two slots for plants. The habitat of these plants can be altered with embedded actuators such as fans, LEDs, and a water pump. The Actuator KP, located in the T-Engine Teaboard2-embedded platform [45], is responsible for controlling these physical actuators. To advertise the functionality it provides, the Actuator KP publishes semantic descriptions about the actuators, as specified in the ACT-ONT, to the SIB. Then it subscribes to the status of these actuators. This way it is aware when the state of the physical actuators needs to be modified. Each of the physical actuators is tagged with an RFID-based ucode tag which is used also as the URI for the given Actuator class instance in the SIB. In addition to providing a natural way to identify physical objects, this approach enables the gardener to modify the semantic data related to the actuators and thus control the physical actuators by touching them with his/her personal device.

Context information about the environmental state such as the humidity, luminosity, and temperature is published by the Sensor KP, located in a Stargate NetBridge sensor platform. The semantic data published by the Sensor KP conforms the sensor ontology presented in section A. Similar to actuators are the sensors also tagged with RFID-based ucode tags. This enables the gardener to read sensor measurements by touching the physical sensors with the ubiquitous communicator.

In addition to the manual control, it is possible to give the responsibility of the greenhouse to an autonomous agent, called autocontrol KP. The auto-control KP, located on an embedded Linux-based Gumstix verdex platform, utilizes information about the sensor measurements and plant preferences to update new state values for the actuators available in the SIB.

6.3. ECSE: Configuration Tool for Smart Spaces. As the name implies event-based configuration of smart environments, ECSE is a tool that enables users to modify semantic technology empowered smart spaces. The ECSE tool [46] was developed to demonstrate and validate our approach for modifying the behavior of semantic technology-based smart spaces (presented in the Section 5.3). The ECSE tool is implemented with Qt framework which allows porting the same software to various computing platforms. The ECSE tool consists of the following separate modules: rule creator, RDFS browser, event/action creator, and rule processor.

The rule creator module is the main component of the ECSE-tool as it provides interface for the user to create new rules to smart spaces. The module utilizes the semantic descriptions of the events and actions in the SIB to present the functionality provided by the environment. Figure 5 illustrates the rule creator view on Linux laptop. In this figure, the ECSE tool is used in the Smart Greenhouse and the gardener creates a rule where the fans of the greenhouse are turned on when the humidity is above 75% and the temperature is above 30.5 Celsius degree.

The RDFS browser module provides a view to the RDF data in the SIB. When combined with the event/action creator module, it is useful for creating new events and actions, even at system run-time. This way it is possible to utilize information about devices which are not familiar with our approach and do not utilize the EA-ONT presented in Section 5.3. It should be noted, however, that this approach needs an expert who first identifies the events and actions from the RDF graph and then assigns them with appropriate ranges and human readable descriptions.

The rule processor module implements the Rule Handler entity of the system model presented in Section 5.3. To demonstrate the whole approach with a single tool, we decided to implement all the necessary components into the ECSE tool. As previously stated the functionality of the Rule Processor is quite simple. It subscribes to the input condition and when necessary executes the actions by updating RDF data to the SIB. For example, with the rule presented in Figure 5, the rule processor would execute the following WQL subscriptions:

WQL(se:HumidityMeasurement, seq(inv(rdf:type), se:hasValue))
WQL(se:TemperatureMeasurement, seq(inv(rdf:type), se:hasValue)).

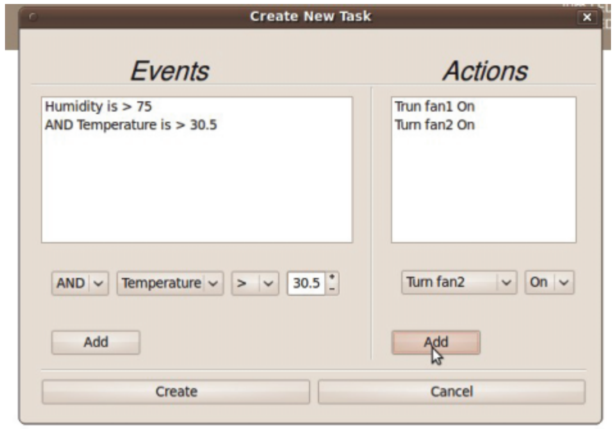


Figure 5: Rule creator view.

This way the rule processor will be aware when the humidity or the temperature changes and can check whether the input condition matches. Once the input condition evaluates as true, the Rule Processor updates information to the SIB as specified in the RDF triple pattern of the actions. For example, in the case of the rule presented in Figure 5, the Rule Handler executes the following *update* operation:

@prefix ac: <http://example.org/actuator/>.
@prefix atag: <ucode:4252:2534:1132:>.
REMOVE(atag:4413 ac:hasStatus "Off".
atag:3211 ac:hasStatus "Off".)
INSERT(atag:4413 ac:hasStatus "On".
atag:3211 ac:hasStatus "On".),

where "4252:2534:1132:1234" is the ucode of fan 1 and "4252:2534:1132:2345" is the ucode of fan 2. When this data is published to the SIB of the Smart Greenhouse, the Actuator KP will be notified about the new status and will modify the physical actuators accordingly.

6.4. Smart Meeting. The Smart Meeting is social smart space application that demonstrated how our approach for autonomous content sharing can be utilized in semantic technology empowered smart spaces [47]. It consists of various autonomous meeting agents that we have implemented to the following smart phones: Nokia N900, Nokia N97, Apple iPhone, and Google Nexus One. These meeting agents enable users to create and participate to meetings and share their contact information and documents with each other.

Smart Meeting imports and extends the FS-ONT presented in Section 5.4. We have defined new classes for the meeting and participant concepts and introduced new subclasses for the existing *File* class. The *Meeting* class presents necessary information about the available meetings in the smart space. The *Participant* class is a subclass of *foaf:Person* class [48] and it represents the contact information of meeting participants. Figure 6 presents how the meeting KPs in the smart phones share information about meetings, participants, files, and file sharing methods via the SIB.

The functionality of the Smart Meeting is best explained via the following scenario. A number of people are waiting

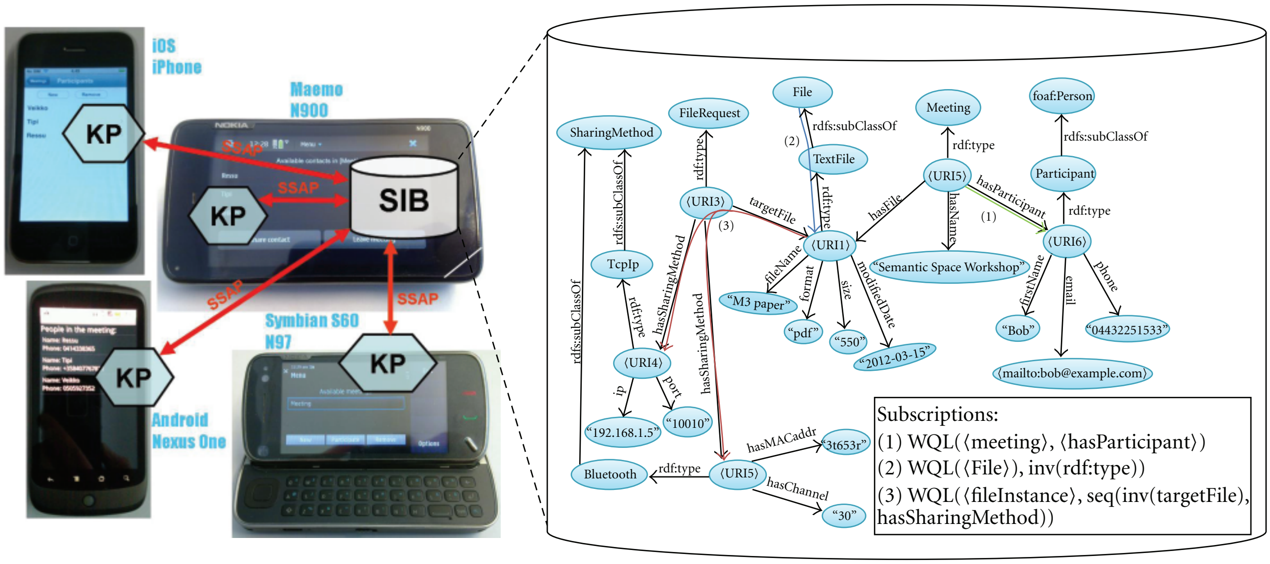


Figure 6: Meeting demonstration set up.

for a workshop on semantic technology empowered smart spaces to begin. One of them is Bob who is going to give a presentation about M3-based semantic interoperability solutions. Before the workshop begins, the session chair commands his meeting agent to set up a smart space and a virtual meeting to the WLAN network of the workshop. Once the smart space and the meeting have been set up, other people whose meeting KPs are connected to the WLAN network of the workshop are informed about the smart space and the meeting. When they join the meeting, their meeting KPs subscribe to the available contact and file information (see subscriptions 1 and 2 in Figure 6).

To advertise his research, Bob requests his meeting KP to publish his contact information and metadata about his scientific paper to the meeting. To do so, the meeting KP inserts the following RDF triples into the SIB:

```
@prefix fi: <http://example.org/fileSharing/>.
@prefix me: <http://example.org/meeting/>.

:bob rdf:type me:participant;
    me:firstName "Bob";
    me:phone "04432251533";
    me:email <mailto:bob@example.com>.
<exampleMeeting> me:hasParticipant:bob.

me:textFile rdfs:subClassOf fi:File.
:file rdf:type me:textFile;
    fi:filename "M3 paper";
    fi:format "pdf";
    fi:size "550";
    fi:modifiedDate "2012-03-15";
<exampleMeeting> me:hasFile:file.
```

Then to be aware when other KPs request the file, Bob's meeting KP subscribes to the sharing methods supported by the requesting KPs as presented in Section 5.4.

When Bob gives his presentation, the chairman takes a picture of him and requests his meeting agent to publish metadata about the picture to the virtual meeting. Soon after Bob's presentation, Mary arrives to the workshop and connects her meeting agent to the smart space. After joining the meeting, Mary's meeting KP subscribes to the available files and informs her about Bob's paper. Mary is interested about the paper and asks her meeting agent to fetch it for her. To request the paper, Mary's meeting KP publishes the following RDF triples into the SIB:

```
@prefix fi: <http://example.org/fileSharing/>.
:request rdf:type fi:fileRequest;
    fi:targetFile <m3paper>;
    fi:hasSharingMethod:bluetooth,:tcp.
:bluetooth rdf:type fi:bluetooth;
    fi:macAddr "3t635r";
    fi:channel "30".
:tcp rdf:type fi:tcp;
    fi:ip "192.168.1.5";
    fi:port "10010",
```

where *m3paper* is the URI of Bob's paper. The SIB notifies Bob's meeting KP about the request made by Mary's meeting KP (see subscription 3 in Figure 6). Similar to Mary's meeting KP, Bob's meeting KP supports both Bluetooth and TCP/IP-based communication and decides to save power by using Bluetooth for transferring the file. After Mary has read the paper, she would be interested to talk with the author of the paper about some interesting issues, but she does not know who the author is. She decides to ask help from her meeting KP who is fortunately able to fetch Bob's contact information and picture based on the linked semantic metadata in the SIB.

7. Conclusions

In this paper, we have described various approaches to support the exploitation of semantic technologies in context-aware smart space applications. When combined, the approaches presented in the paper are very important because they enable the creation of pervasive computing systems which can take full advantage of the semantic technologies. The developed approaches have been built according to the principles of M3 concept and validated with prototype smart space applications and tools.

First, we described approaches and methods for tackling the challenges of implementing a semantic technology empowered smart space agent in a resource-restricted low-power platform. The approach was validated by implementing autonomous agent called Active Tag with total code size of 39.7 kB and RAM consumption of 25.3 kB. Second, we described a uID-based method usable for identifying physical objects and locating information related to these objects from all over the world. This approach was demonstrated both in the Smart Home Garden, and Smart Greenhouse smart spaces. Third, we presented a novel approach that allows the end-users to modify the functionality provided by smart spaces. The main aspect of the approach is that it provides the user with natural human language description of the smart space capabilities so that the user is able to define what kind of functionality is required from the smart space. The approach was realized in practice by implementing a general-purpose ECSE tool with Qt-framework. Fourth, we described how content such as video, audio, and documents that are not feasible to be stored in the semantic database as such can be shared between the semantic technology empowered autonomous agents. The Smart Meeting application demonstrated this approach in practice.

By utilizing the approaches and methods presented in this paper, it will be possible to create context-aware applications for semantic technology empowered smart spaces. However, there is still a lot of work to be done before semantic technology-based context presentation can be utilized in large-scale real-life smart spaces. For example, security and privacy issues are important, and although the RIBS already provides some level of security there is still a lot to be done in this field. Other interesting future research areas in semantic technology empowered smart spaces include, for example, data maintenance and ontology governance.

Acknowledgment

This work has been funded by the Device and Interoperable Ecosystem (DIEM) and the Merging IoT Technologies (MIoTE) Projects.

References

[1] A. K. Dey, *Providing architectural support for building context-aware applications [Ph.D. thesis]*, Georgia Institute of Technology, 2000.

[2] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–100, 1991.

[3] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *Computer*, vol. 36, no. 3, pp. 25–31, 2003.

[4] E. Aarts, H. Harwing, and M. Schuurmans, "Ambient intelligence," in *The Invisible Future*, J. Denning, Ed., pp. 235–250, McGraw Hill, New York, NY, USA, 2001.

[5] N. Gershenfeld, "The internet of things," *Scientific American*, vol. 291, no. 4, 2004.

[6] J. R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine, and W. Buxton, "Evolution of a reactive environment," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI '95)*, pp. 170–177, Denver, Colo, USA, May 1995.

[7] M. Dertouzos, *The Future of Computing*, Scientific American, 1999.

[8] B. Brumitt, B. Myers, J. Krum, A. Kern, and S. Shafer, "Easy-Living: technologies for intelligent environments," in *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC '00)*, vol. 1927 of *Lecture Notes in Computer Science*, pp. 12–29, Springer, 2000.

[9] T. Kindberg, J. Barton, J. Morgan et al., "People, places, things: web presence for the real world," in *Proceedings of the 3rd IEEE Workshop Mobile Computing Systems and Applications (WMCSA '00)*, p. 19, IEEE CS Press, 2000.

[10] B. Johanson, A. Fox, and T. Winograd, "The interactive workspaces project: experiences with ubiquitous computing rooms," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 67–74, 2002.

[11] Contributing Members of UPnP Forum, "UPnP device architecture 1. 1," http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

[12] OSGi Alliance web page, "OSGi—the dynamic module system for Java," http://www.osgi.org/.

[13] M. Gudgin, M. Hadley, N. Mendelsohn et al., *SOAP Version 1. 2 Part 1: Messaging Framework*, W3C Recommendation, 2nd edition, 2007.

[14] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," in *Proceedings of the International Conference on Software Engineering (ICSE '00)*, pp. 407–416, June 2000.

[15] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[16] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data—the story so far," *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[17] H. Chen, *An intelligent broker architecture for pervasive and context-aware systems [doctoral dissertation]*, University of Maryland, Department of Computer Science and Electrical Engineering, Baltimore County, Md, USA, 2004.

[18] X. Wang, J. S. Dong, C. Chin, S. R. Hettiarachchi, and D. Zhang, "Semantic space: an infrastructure for smart spaces," *IEEE Pervasive Computing*, vol. 3, no. 3, pp. 32–39, 2004.

[19] G. Klyne and J. J. Carroll, *Resource Description Framework (RDF): Concepts and Abstract Syntax*, W3C Recommendation, 2004.

[20] D. Brickley and R. V. Guha, *RDF Vocabulary Description Language 1. 0: RDF Schema*, W3C Recommendation, 2004.

[21] W3C OWL Working Group, *OWL 2 Web Ontology Language Document Overview*, W3C Recommendation, 2009.

[22] R. Masuoka, B. Parsia, and Y. Labrou, "Task computing—the semantic web meets pervasive computing," in *Proceedings of the 2nd International Semantic Web Conference (ISWC '03)*, pp. 886–881, 2003.

[23] S. Ben Mokhtar, N. Georgantas, and V. Issarny, "COCOA: conversation-based service composition in pervasive computing environments with QoS support," *Journal of Systems and Software*, vol. 80, no. 12, pp. 1941–1955, 2007.

[24] Z. Song, A. A. Cárdenas, and R. Masuoka, "Semantic middleware for the internet of things," in *Proceedings of the 2nd International Internet of Things Conference (IoT '10)*, pp. 1–8, December 2010.

[25] G. Thomson, S. Bianco, S. Mokhtar, N. Georgantas, and V. Issarny, "Amigo aware services," in *Communications in Computer and Information Science*, vol. 11, pp. 385–390, 2008.

[26] P. Liuha, J. Soininen, and R. Otaolea, "SOFIA: opening embedded information for smart applications," in *Proceedings of the Embedded World Conference*, Nuremberg, Germany, March 2010.

[27] M. Compton, P. Barnaghi, L. Bermudez et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[28] H. Chen, F. Perich, T. Finin, and A. Joshi, "SOUPA: standard ontology for ubiquitous and pervasive applications," in *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS '04)*, pp. 258–267, August 2004.

[29] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, "Smart-M3 information sharing platform," in *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC '10)*, pp. 1041–1046, Riccione, Italy, June 2010.

[30] J. Suomalainen, P. Hyttinen, and P. Tarvainen, "Secure information sharing between heterogeneous embedded devices," in *Proceedings of the 4th European Conference on Software Architecture: Doctoral Symposium, Industrial Track and Workshops (ECSA '10)*, pp. 205–212, August 2010.

[31] A. Lappeteläinen, J. Tuupola, A. Palin, and T. Eriksson, "Networked systems, services and information—the ultimate digital convergence," in *Proceedings of the 1st International Conference on Network on Terminal Architecture (NoTA '08)*, pp. 1–7, 2008.

[32] D. Manzaroli, L. Roffia, T. S. Cinotti et al., "Smart-M3 and OSGi: the interoperability platform," in *Proceedings of the International Workshop on Semantic Interoperability for Smart Spaces (SISS '10)*, pp. 1053–1058, IEEE Press, June 2010.

[33] O. Lassila, *Programming semantic web applications: a synthesis of knowledge representation and semi-structured data [doctoral dissertation]*, Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Software Technology, 2007.

[34] E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language For RDF*, W3C Recommendation, 2008.

[35] The OWL Services Coalition, "OWL-S: semantic markup for web services," 2003, http://www.daml.org/services/owl-s/1.0/owl-s.html.

[36] T. Berners-Lee, R. Fielding, and L. Masinter, *Uniform Resource Identifiers (URI): Generic Syntax*, Internet Draft Standard RFC, 2396, IETF, 1998.

[37] M. Duerst and M. Suignard, *Internationalized Resource Identifiers (IRIs)*, Proposed Standard RFC, 3987, IETF, 2005.

[38] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *Knowledge Engineering Review*, vol. 18, no. 3, pp. 197–207, 2003.

[39] K. Hansen, W. Zang, J. Fernandes, and M. Ingstrup, "Semantic web ontologies for ambient intelligence," in *Proceedings of the 1st International Research Workshop on The Internet of Things and Services*, 2008.

[40] N. Koshizuka and K. Sakamura, "Ubiquitous ID: standards for ubiquitous computing and the internet of things," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 98–101, 2010.

[41] D. Beckett and T. Berners-Lee, *Turtle—Terse RDF Triple Language*, W3C Team Submission, 2011.

[42] A. Ylisaukko-oja, P. Hyttinen, J. Kiljander, J. Soininen, and E. Viljamaa, "Semantic inteface for resource restricted wireless sensors," in *Proceedings of the IC3K 2nd International Workshop on Semantic Sensor Web (SSW '11)*, Paris, France, October, 2011.

[43] J. Kiljander, M. Eteläperä, J. Takalo-Mattila, and J. P. Soininen, "Opening information of low capacity embedded systems for Smart Spaces," in *Proceedings of the 8th IEEE Workshop on Intelligent Solutions in Embedded Systems (WISES '10)*, pp. 23–28, July 2010.

[44] J. Takalo-Mattila, J. Kiljander, M. Eteläperä, and J. Soininen, "Ubiquitous computing by utilizing semantic interoperability with item-level object identification," in *Proceedings of the 2nd International ICST Conference on Mobile Networks and Management (MONAMI '10)*, vol. 68, pp. 198–209, Springer, 2010.

[45] K. Sakamura and N. Koshizuka, "T-engine: the open, real-time embedded-systems platform," *IEEE Micro*, vol. 22, no. 6, pp. 48–57, 2002.

[46] J. Kiljander, J. Takalo-Mattila, M. Eteläperä, K. Keinänen, and J. Soininen, "Enabling end-users to configure smart environments," in *Proceedings of the International Symposium on Applications and the Internet (SAINT '11)*, pp. 303–308, 2011.

[47] J. Kiljander, M. Eteläperä, J. Takalo-Mattila, K. Keinänen, and J. Soininen, "Autonomous file sharing for smart environments," in *Proceedings of the International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS '11)*, pp. 191–196, 2011.

[48] D. Brickley and L. Miller, *FOAF Vocabulary Specification 0. 98*, Namespace Document, 2010.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 790374, 10 pages
doi:10.1155/2012/790374

Research Article

A Fusion Approach of RSSI and LQI for Indoor Localization System Using Adaptive Smoothers

Sharly Joana Halder and Wooju Kim

Department of Information and Industrial Engineering, College of Engineering, Yonsei University, Seoul 120-749, Republic of Korea

Correspondence should be addressed to Sharly Joana Halder, sjhalder@yonsei.ac.kr

Received 21 February 2012; Revised 8 August 2012; Accepted 30 August 2012

Academic Editor: MoonBae Song

Due to the ease of development and inexpensiveness, indoor localization systems are getting a significant attention but, with recent advancement in context and location aware technologies, the solutions for indoor tracking and localization had become more critical. Ranging methods play a basic role in the localization system, in which received signal strength indicator- (RSSI-) based ranging technique gets the most attraction. To predict the position of an unknown node, RSSI measurement is an easy and reliable method for distance estimation. In indoor environments, the accuracy of the RSSI-based localization method is affected by strong variation, specially often containing substantial amounts of metal and other such reflective materials that affect the propagation of radio-frequency signals in nontrivial ways, causing multipath effects, dead spots, noise, and interference. This paper proposes an adaptive smoother based location and tracking algorithm for indoor positioning by making fusion of RSSI and link quality indicator (LQI), which is particularly well suited to support context aware computing. The experimental results showed that the proposed mathematical method can reduce the average error around 25%, and it is always better than the other existing interference avoidance algorithms.

1. Introduction

Due to the ease in deployment, inexpensiveness and potential applications in the smart building, security, and healthcare, indoor localization system gets significant attentions in these recent years. Mobile positioning has become increasingly interesting system most notably for context-aware application and emergency services, which works in ad hoc manner. Global positioning system (GPS) has been the mainstream technology for location and tracking for outdoor environment but the positioning within indoor environments does not permit a positioning with GPS (or only with poor quality). It generally requires a direct view to several satellites, resulting in limited performance for indoor environments. GPS signal in an indoor environment is too faint to provide sufficient accuracy. Development of non-GPS-based solutions is thus of great interest for indoor use based on existing signals and hardware, as well as new systems and sensor modalities.

A number of commercial systems and research prototypes are developed with different kinds of localization methods. This method usually use infrared (IR) [1], ultrasound [2–4], or radio-frequency (RF) system [5–7]. Different sensors provide a different range of accuracy from centimeters to room level. It seems as if the accuracy is smaller than a room. But in practice, when we directly transform (x, y) coordinates to room level information, it often causes mistakes. The reason is that the wireless signal is easily disturbed, making localization unusual jumping in a split second or over a short span, and such situations may affect the location estimation from one room to another. Infrared system has been popularly used for containment-based location systems [1]. The response speed of infrared transducers allows a fair amount of data to be transmitted.

Although localizing techniques using ultrasound have been actively researched because ultrasound has a number of merits such as low cost and relative easiness of application; but, ultrasound is still considered as a low-guaranteed sensor since it is too sensitive to be used for precise localization. RF-based positioning systems are probably the most popular for indoor tracking. But there is a significant amount of time that the localization accuracy may suffer from the bursts of background noise [8]. The reason was investigated by recent experimental studies [9], which show that the various indoor fading effects are the main causations of the localization error [10]. The fading effects can be divided into the slow fadings and the fast fadings. Slow fadings are effects caused by the environment, such as the radio block or shadow fading. Fast fading is temporary or random effect, such as the interference, random noise, and the multipath effects. They work together making the RSSI value hard to predict the distance, thus introducing a large localization error. Current localization systems use different methods to improve the localization accuracy.

Mainly these algorithms are categorized into two—signature based and beacon based. Most indoor localization systems employ an RSSI-signature-based approach which exploits temporal stability in the RSSI of wireless signals. In that, at every known location, the RSSIs collected from a set of predeployed beacons form an RSSI signature for the corresponding location. When a target carrying a receiving tag enters the space, the RSSI values collected on the tag are compared to the RSSI signatures. The location of the target is identified by the corresponding location with the closest RSSI signature. To tackle the temporal variation of RSSI signatures, methods of ensuring robust mapping between the measured RSSI values and the prerecorded RSSI signatures have been studied intensively in recent years [11]. *RADAR* was one of the first RF signal strength-based positioning system used to track people inside buildings [6]. The major disadvantages of the fingerprinting method include the need for dense training coverage and poor extrapolation to areas not covered during training.

In contrast to fingerprinting, model-based positioning techniques express the RF signal attenuation using physics-based "path loss" model [12, 13]. From an observed RSSI, these methods triangulate the person based on a distance calculation from multiple access points. However, the position-RSSI relationship is highly complex due to multipath, metal reflection, and interference noise. Thus, the RSSI propagation may not be adequately captured by a fixed invariant model.

In this paper, the proposed protocol improves the existing algorithms using RSSI and LQI values. RSSI and LQI [13] are considered as two parameters which play a pivotal role in the beacon-based localization of sensor nodes. Typically RSSI is a measure of dBm, which is ten times the logarithm of the ratio of the power (P) at the receiving end and the reference power (P_{ref}). Power at the receiving end is inversely proportional to the square of distance.

Hence, RSSI could potentially be used as an indicator of the distance at which the sending mote is located from the receiving mote. When data from many such neighboring motes are combined, the location of the sending mote can be judged with reasonable accuracy. The localization systems presented in this paper are based on the RSSI as a strength indicator and LQI as a quality indicator of a received packet and it can also be used to estimate a distance from nodes to reference points. The LQI has been used as an assistant indicator of RSSI indicator. The proposed protocol provides a filtering process of an object based on its distance.

Table 1: Low and high RF range of LQI and RSSI.

	Low RF	High RF
LQI	105	108
RSSI	−75 dBm	−25 dBm

LQI exhibits a very good correlation with packet loss and is therefore a good link quality indicator. However, one of the contributions of the present work is to show that RSSI is a reasonable metric if it is processed correctly, and if interference can be distinguished from noise. Given that LQI is a superior metric, it should not be forgotten it is only made available by 802.15.4-compliant devices. It therefore makes sense to make the most out of RSSI. Low and high RF range of LQI and RSSI is depicted in Table 1.

In our paper we use an adaptive filter as it performs well to track an object under such changing conditions in the RF signal environment. In this paper, the proposed protocol tries to improve the existing algorithms [5, 13], using RSSI and LQI values. The indoor localization systems presented in this report are based on the RSSI as a strength indicator and LQI as a quality indicator of received packets. It can also be used to estimate a distance from a node to reference points. This system uses the LQI and RSSI in a different way and therefore it could lead to better and more predictable results than the other existing system. Several experiments were conducted to investigate the performance of the proposed scheme. Firstly, this system performs with respect to the signal analysis to understand the characteristic of the LQI and RSSI values on three types of environments to decide how the environment effects on RSSI and LQI strength. The effect of distance on received signal strength (RSS) can be measured by RSSI and LQI provided by the radio. Secondly, this scheme performs with respect to the signal analysis which is to filter the original signals in order to remove the noise. Besides, the noise could be estimated by using adaptive filtering algorithms. Sudden peaks and gaps in the signal strength are removed and the whole signal is smoothed, which eases the analysis process. We propose three different types of new filtering to smooth the real RSSI, that is, "LQI" filtering, fusion filtering, and "BOTH" filtering, and compare the results.

The remainder of this paper consists of six subsections. Section 2 describes brief explanation of some properties of RSSI, LQI, and two simple filters which could be used to smooth the RSSI values. In addition, it also explains the proposed localization algorithm briefly. Section 3 reveals the experimental testbed. The system implementation of "A Fusion Approach of RSSI and LQI for Indoor Localization System using Adaptive Smoothers" and its probability of returning the correct location are explained in Section 4. Section 5 compares the experimental performance. And Section 6 concludes the section with conclusions.

2. System Configuration

This section gives a brief explanation about RSSI, LQI, and two common filters, which are simple averaging and feedback filters and then will focus on how this effective protocol has been implemented, and implementation issues were considered. To measure the radio strength, two useful radio hardware link quality metrics were used in this experiment, that is, (i) LQI and (ii) RSSI. Specifically, RSSI is the estimation of the signal power and is calculated over 8 symbol periods, while LQI can be viewed as chip error rate and is calculated over 8 symbols following the start frame delimiter (SFD). The specific point in a system where position estimates are calculated is an important design parameter. In this scheme, the mobile device itself calculates the position. The device calculates its own position based on its own measurements.

2.1. Received Signal Strength Indicator (RSSI). Majority of the existing methods leverage the existence of IEEE 802.11 base stations with powerful radio transmit powers of approximately 100 mW per base station. Such radios are in a different class from the low power IEEE 802.15.4 compliant radios that typically transmit at low power levels ranging from 52 mW to 29 mW. The wide availability of larger number of IEEE 802.15.4 radios has revived the interest for signal strength-based localization in sensor network. Despite of rapidly increasing popularity of IEEE 802.15.4 radios and signal strength localization, there is a lack of detailed characterization of the fundamental factors contributing to large signal strength variation. The analysis of RSSI values is needed to understand the underlying features of location-dependent RSSI patterns and location fingerprints. An understanding of the properties of the RSSI values for location can assist in improving the design of positioning algorithms and in deployment of indoor positioning systems. The characteristics of RSS will decrease with increased distance as the equation shows below:

$$\mathrm{RSSI} = -(10n\log 10\, d + A), \tag{1}$$

where, n: signal propagation constant, also named propagation exponent; d: distance from sender; A: received signal strength at a distance of one meter.

Lots of localization algorithms require a distance to estimate the position of unknown devices. One possibility to acquire a distance is measuring the RSS of the incoming radio signal. The idea behind RSS is that the configured transmission power at the transmitting device (PTX) directly affects the receiving power at the receiving device (PRX). According to Friis' free space transmission equation, the detected signal strength decreases quadratically with the distance to the sender (Figure 1(a)):

$$\mathrm{PRX} = \mathrm{PTX} * \mathrm{GTX} * \mathrm{GRX}\left(\frac{\lambda}{4\pi d}\right)^2, \tag{2}$$

where, PTX: transmission power of sender, PRX: remaining power of wave at receiver, GTX: gain of transmitter,

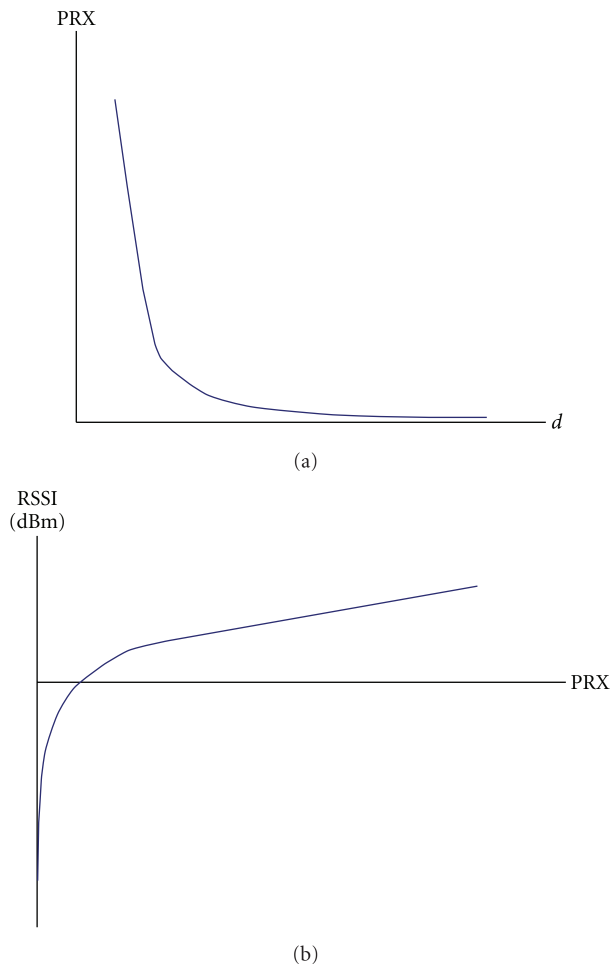


Figure 1: (a) Received power PRX versus distance to the transmitter. (b) RSSI as quality identifier of the received signal power PRX.

GRX: gain of receiver, λ: wave length, and d: distance between sender and receiver.

In embedded devices, the RSS is converted to RSSI which is defined as ratio of the received power to the reference power (P_{Ref}). Typically, the reference power represents an absolute value of $P_{\mathrm{ref}} = 1$ mW.

$$\mathrm{RSSI} = 10 * \log\frac{\mathrm{PRX}}{\mathrm{PRF}}[\mathrm{RSSI}] = \mathrm{dBm}. \tag{3}$$

An increasing received power results a rising RSSI. Figure 1(b) illustrates the relation between RSSI and the received signal power. Plotting RSSI versus distance "d" results in a graph, which is in principle axis symmetric to the abscissa. Thus, distance "d" is indirect proportional to RSSI. In practical scenarios, the ideal distribution of PRX is not applicable, because the propagation of the radio signal is interfered with a lot of influencing effects.

2.2. Link Quality Indicator (LQI). For communications IEEE 802.15.4 radios provide applications with information about the incoming signal [14]. The effect of distance on RSS can be measured by the packet success rate, RSSI, and LQI provided by the radio. LQI is a metric introduced in IEEE 802.15.4 that measures the error in the incoming modulation of successfully received packets (packets that pass the CRC criterion). The LQI metric characterizes the strength and quality of a received packet. It is introduced in the 802.15.4 standard [15] and is provided by CC2430 [14]. LQI measures each successfully received packet and the resulting integer ranges from 0×00 to $0 \times \text{ff}$ (0–255), indicating the lowest and highest quality signals detectable by the receiver (between -100 dBm and 0 dBm). The correlation value of LQI range from 50 to 110 where 50 indicates the minimum value and represents the maximum. The 50 is typically the lowest quality frames detectable by CC2430. Software must convert the correlation value to the range 0–255, for example, by calculating

$$\text{LQI} = (\text{CORR} - a) \cdot b, \tag{4}$$

where, CORR: correlation value and a and b are found empirically.

The CORR (correlation value) is the raw LQI value which can be obtained from the last byte of the message. The raw value can be gotten from CC2430 (CORR) which is between 40 and 110. Limited to the range 0–255, "a" and "b" are found empirically based on PER measurements as a function of the correlation value. A combination of RSSI and correlation values may also be used to generate the LQI value. LQI values are uniformly distributed between these two limits. Different form RSSI, LQI measures the qualities of links while RSSI measures the strengths of links. LQI is a measure of the error in the signal, not the strength of the signal. A "weak" signal may still be a very crisp signal with no errors and thus a potentially good routing neighbor. If there is no interference from other 2.4 GHz devices, then LQI will generally be good over distance. Note that scaling the link quality to an LQI, compliant with IEEE 802.15.4, must be done by software. This can be done on the basis of the RSSI value, the correlation value, or a combination of those two. Signal strength and link quality values are not necessarily linked. But if the LQI is low, it is more likely that the RSSI will be low as well. Nevertheless, they also depend on the emitting power. Even though they do not describe how far from each other the sender and the receiver are located, it illustrates perfectly that both low and high power emissions guarantee a good link quality. The low RF emissions could be more sensitive to external disturbances. LQI exhibits a very good correlation with packet loss and is therefore a good link quality indicator. However, one of the contributions of the present work is to show that RSSI is a reasonable metric if it is processed correctly, and if interference can be distinguished from noise. Given that LQI is a superior metric, it should not be forgotten that it is only made available by 802.15.4-compliant devices. It therefore makes sense to make the most out of RSSI.

Various filters can be used to smooth the RSSI value [7, 8, 16, 17]. Two common filters are simple averaging and feedback filters. Averaging is the most basic filter type, but it requires more data packets to be sent. Feedback filters use only a small part of the most recent RSSI value for each calculation. This requires less data, but increases the latency when calculating a new position.

2.2.1. Averaging Filter. The average RSSI value is simply calculated by requiring a few packets from each reference node, each time the RSSI values are measured and calculated according to the equation below:

$$\overline{\text{RSSI}} = \frac{1}{n}\sum_{i=0}^{i=n}\text{RSSI}_i. \tag{5}$$

2.2.2. Feedback Filter. If a filter approximation is used, this can be done as shown below. In this equation the variable $a$is typically 0.75 or above. This approach ensures that a large difference in RSSI values will be smoothed. Therefore, it is not advisable if the assets that should be tracked can move long distances between each calculation:

$$\text{RSSI}n = a * \text{RSSI}n + (1 - a) * \text{RSSI}n - 1. \tag{6}$$

This means that the averaged RSSI value corresponding to the signal strength at distance depends on both the previous averaged value and the most recently measured value. As the value of "a", which should be between zero and one, determines the degree of filtering if "a" is chosen to be close to one; the new measurement barely plays a role in the calculation of the new average. If on the other hand the value of "a" is nearly zero, virtually no filtering is performed. An optimal filter, that is, a value for "a", specifically for this project will be determined in this section. In this paper we are going to propose three filtering processes. At first, we use an LQI filter to smooth the RSSI value. In our second filtering process, we proposed a fusion filter which is the fusion of RSSI and LQI where we decided to use LQI as a reference aid when the RSSI or LQI is below RSSI and LQI threshold. Lastly, we proposed a BOTH filter, which is also a fusion of RSSI and LQI where we decided to use LQI as a reference aid when the RSSI or LQI is below RSSI or LQI threshold.

2.3. Localization Algorithm. This scheme decides to use a private and scalable system. It features an active base station that transmits both RSSI and LQI signals. The mobile devices receive the signals, but they do not transmit anything themselves. The base station transmits the RSSI and LQI signals at the same moment in time. A mobile device measures signal and is able to calculate the distance to the transmitter. By this scheme the location privacy of the user, who carries the mobile device, can be easily guaranteed because the mobile device does not send out any signals that might disclose its presence or its location. A further advantage of this architecture is scalability to many mobile devices. As the mobile devices do not transmit any signals, there can be an unlimited number of mobile devices in principle. Due to its privacy and scalability features, this architecture might be particularly suitable for large-scale professional location systems or systems in public spaces. Each mobile device calculates its own position, based on the received signals.

As we know, for environmental change the log model also changes, so the proposed system uses a scaling factor for adjusting the log model with the measured data. This system includes a scaling factor "s" with the basic RSSI log model equation, which adjusted the log model based on the received signal:

$$\mathrm{RSSI} = -10n\log_{10}(sd+1) + A. \quad (7)$$

To get the best performance of filtered data, this scheme uses triplicate filtering process for smoothing the received RSSI data. To do this experiment, we determine the value of filtering factor "a" and applied that value to the existing feedback filtering equation. To filter the received RF signal, firstly we applied the LQI filter based on the received packet quality. For our second experiment we used a fusion of LQI and RSSI values based on the filtering conditions to smooth the measured RSSI. There is a detail explanation in section IV-B. BOTH filter is our last experiment, where we filter the received RSSI values with both LQI and RSSI values. Section IV-C gives a detail explanation about this filtering process.

We apply the LQI filtering by using the following equation:

$$\mathrm{smooth\_RSSI}_{t(\mathrm{LQI})} = a * \mathrm{RSSI}_t + (1-a) * \mathrm{RSSI}_{t-1}. \quad (8)$$

Secondly, we proposed a fusion filtering of RSSI and LQI values, for smoothing the measured RSSI:

$$\mathrm{smooth\_RSSI}_{t(\mathrm{Fusion})} = a * \mathrm{RSSI}_t + (1-a) * \mathrm{RSSI}_{t-1}. \quad (9)$$

At last we proposed a BOTH filtering of RSSI and LQI values, for smoothing the measured RSSI. Then we compared the resulted values of these three filters. We, however, used the following equation for smoothing the measured RSSI for BOTH filter:

$$\mathrm{smooth\_RSSI}_{t(\mathrm{BOTH})} = \mathrm{a} * \mathrm{RSSI}_t + (1-\mathrm{a}) * \mathrm{RSSI}_t - 1. \quad (10)$$

3. Experimental Testbed

Adaptive filter contains a set of adjustable parameters. In design problem the requirement is to find the optimum set of filter parameters from knowledge of relevant signal characteristics according to some criterion. This mathematical system combines the general principles of a proximity-based localization system with the analysis of the radio signal strength behavior over distance. This system uses the LQI and RSSI in a different way and therefore it could lead to better and more predictable results than the other existing system. Several experiments had conducted to investigate the performance of the proposed scheme.

To implement our proposed protocol we decided to use 10 m of an indoor environment as our experimental testbed. We implemented our experiment in three kinds of environments. The first step of this system performs with respect to the signal analysis to understand the characteristic of LQI and RSSI values on three types of environments.

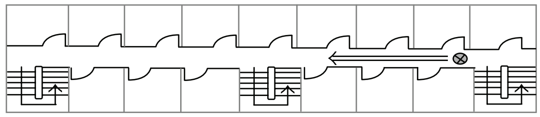


Figure 2: Close space indoor environment (path 1).

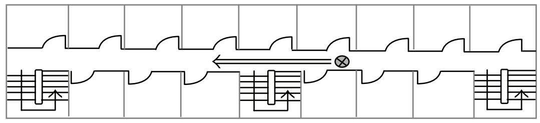


Figure 3: Half-open space indoor environment (path 2).

The effect of distance on RSS can be measured by RSSI and LQI provided by the radio. Equation (1) describes the basic model formula for RSSI where RSS decrease with increased distance, but this scheme decided to use a scaling factor "s" in the basic log model equation to adjust the log model with measured RSSI values. So, to find the accurate log model for specific environment we use a scaling factor "s" in (7). The experiment is conducted on three types of following environment, that is, Figures 2, 3, and 4 to decide how the environment effects RSSI and LQI strength. The first experiment is conducted in close space indoor environment (Figure 1). The second experiment is deployed in half-open space indoor environment, where few meters of the corridor were open (Figure 2) and the third experiment (Figure 3) is conducted on the open space indoor environment to decide the variation of the RF and LQI from the other two experiments.

To determine the accurate distance, the following distance equation has been used:

$$\mathrm{Distance} = \frac{(\mathrm{RSSI}-A)/(10-10n)-1}{s}. \quad (11)$$

And for measuring the signal attenuation factor the following equation has been used:

$$N = \frac{\mathrm{RSSI}-A}{-10\log 10(sd+1)}. \quad (12)$$

Figure 5 represents the measured RSSI and LQI values. These figures are also representing how the log model curve adjusted with the measured RSSI values in half-open space indoor environment.

Figures 6 and 7 provide the reliability of the RSSI and LQI for all testbeds.

Figures 8 and 9 represent the average distance error according to RSSI and LQI values for all three testbeds. Figure 8 shows that in 8 meters the LQI value is 100. From our experiment we found that when LQI = 100, it gives 20% reliability. Whereas, Figure 9 reveals that in 8 meters RSSI value is −77, from our experiment we found that it gives 10% reliability. So, we conclude that over long distance LQI reliability is better than RSSI. According to this finding; we decide to use LQI as an assistance filtering factor for RSSI, which we are going to discuss next.

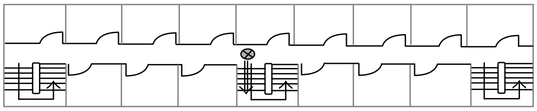


Figure 4: Open space indoor environment (path 3).

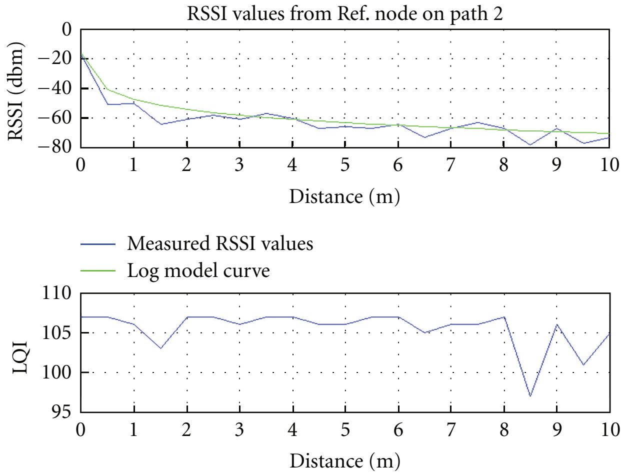


Figure 5: Characteristics of RSSI and LQI in half-open space indoor environment.

It is found that LQI gives best performance when the value was 108 in about 2 meters distance, which indicate that it gives 80% reliability (Figure 8). The results show that when the value is 100, it gives lowest performance in about 8 meters. As our measurement testbed was 10 meters, we decided to determine the value below 10 meters. So, we determined the LQI reliability from 100 to 108, where the reliability varies from 20% to 80%, which means LQI filtering factor "a" varies between 0.8 to 0.2 for LQI filtering. And we also determine that if "a" value is below 101 then it should be negligible. The following equation has been used for LQI filtering:

$$\text{smooth\_RSSI}_{t(\text{LQI})} = a * \text{RSSI}_t + (1 - a) * \text{RSSI}_{t-1},$$
$$a = 0.8 - 0.6 * \frac{108 - \text{LQI}}{8}. \tag{13}$$

Furthermore, RSSI gave best performance when its value was -15 in about 0 meter distances, by which we determine that it gives 100% reliability (Figure 9). It has also been seen that when its value was -75 it gave the lowest performance in about 5 meters, which means 50% reliability. So we decided the reliability for RSSI between -15 to -75, where the reliability varies from 50% to 100%. So this system decided to use the RSSI filtering factor a around 1 to 0.5 according to RSSI value. We also determine that if RSSI value is below -75 then the RSSI reliability will be 10%, which could be

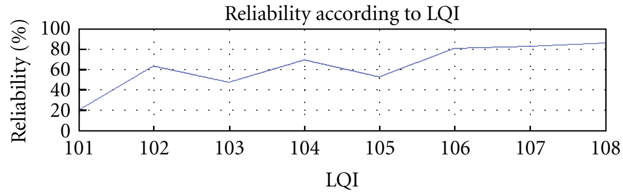


Figure 6: Reliability according to LQI in all three paths.

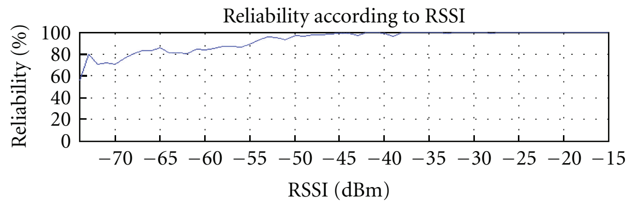


Figure 7: Reliability according to RSSI in all three paths.

neglected. However, the following equation was used for RSSI filtering:

$$\text{smooth\_RSSI}_{t(\text{RSSI})} = a * \text{RSSI}_t + (1 - a) * \text{RSSI}_{t-1},$$
$$a = 1 - 0.5 * \frac{-15 - \text{RSSI}}{60}. \tag{14}$$

The proposed system was implemented with TinyOS 2.x and the ZigBee device (Hybus, Hmote 2430), which were designed to operate in environments where they are approximately coplanar, therefore constrained in 3 of their 6 degrees of freedom (pitch, roll, and z-axis). Hardware had already been developed as a research platform. This section will describe these devices. The performance of the systems has been analysed by implementation and simulation using MATLAB.

4. System Implementation

The proposed scheme performs with respect to the signal analysis to filter the original signals in order to remove the noise. By smoothing, sudden peaks and gaps in the signal strength are removed and the whole signal is smoothed, which eases the analysis process.

4.1. LQI Filter. LQI is a metric, provided by radio. It measures the error in the incoming modulation of successfully received packets. LQI exhibits as a quality indicator of a received packet and helps to estimate the distance between the nodes. For smoothing the measured RSSI values we use LQI filter based on only LQI values. The condition for this filter is as follows: if the LQI value is smaller than 100 then

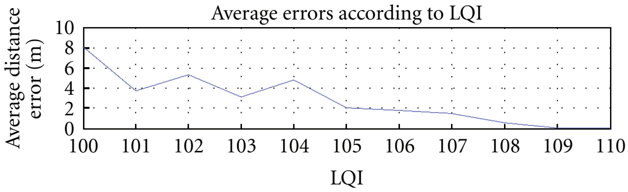


Figure 8: Average errors according to LQI in all three paths.

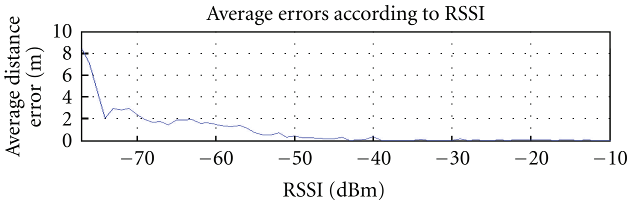


Figure 9: Average errors according to RSSI in all three paths.

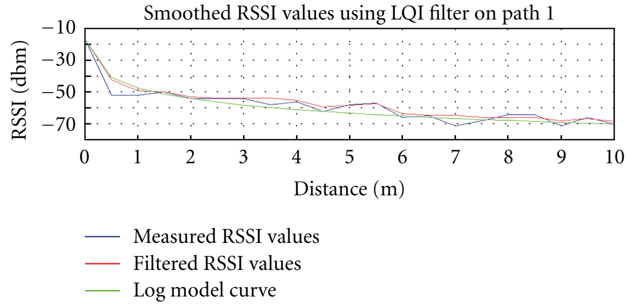


Figure 10: Smoothed RSSI values using LQI filter for path 1.

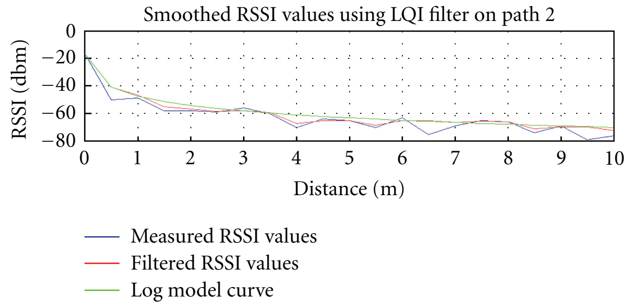


Figure 11: Smoothed RSSI values using LQI filter for path 2.

the filtering factor α will be 0 otherwise the measured RSSI signal will be smoothed by the LQI filter:

$$\begin{aligned}
&\text{if LQI} < 100 \\
&\qquad \alpha = 0 \\
&\text{else} \\
&\text{smooth_RSSI}_{t(\text{LQI})} = a * \text{RSSI}_t + (1 - a) * \text{RSSI}_{t-1}.
\end{aligned} \tag{15}$$

After applying the filter, the program analyses the behavior of the filtered signal strengths over distance. Figures 10, 11, and 12 provide the results. Figures 10, 11, and 12 show the analysis results of LQI filter for 3 different paths. To analyse the filter result, we measured the log model using the scaling factor for adjusting the log model with the measured data. These figures (Figures 10, 11, and 12) show the difference between filtered RSSI value and raw RSSI values and also how the proposed scaling factor helps to adjust the log model based on three different paths. LQI filter gave the best performance in closed indoor environment.

4.2. Fusion Filter. In our second experiment we make a fusion of RSSI and LQI to smooth the measured RSSI values. For this fusion filtering we use the LQI filtering in case of sudden peaks and shaded signals. In this fusion filter we decided to measure the difference between present RSSI value and previous RSSI value. If the difference of present and previous RSSI value is smaller than RSSI threshold value or if the LQI value is smaller than LQI threshold value then the signal will be filtered by RSSI filter otherwise it will use LQI filter. To determine the LQI threshold value we use the defined LQI threshold value 105. But for determining RSSI threshold value we did three kinds of experiments. Firstly we decided to use the smallest peak value (SPV) as an RSSI threshold. Secondly, we averaged the minimum and maximum RSSI values (AMM). And finally we averaged each distance difference of all RSSI values (ADD) and we determined that, by using ADD (average distance difference) value as an RSSI threshold, filtering is happening more smoothly than the other two values:

$$\begin{aligned}
&\text{if}(\text{abs}\,(\text{RSSI}_t - \text{RSSI}_{t-1}) < \text{RSSI}_{\text{Threshold}}) \\
&\qquad \mid (\text{LQIt} < \text{LQI}_{\text{Threshold}}) \\
&\qquad\qquad \text{RSSI filter} \\
&\text{else} \\
&\qquad\qquad \text{LQI filter} \\
&\text{smooth_RSSI}_{(\text{Fusion})} = a * \text{RSSI}_t + (1 - a) * \text{RSSI}_{t-1}.
\end{aligned} \tag{16}$$

Figures 13, 14, and 15 show the results for 3 paths by using the fusion filter. Figures 13, 14, and 15 are showing three types of RSSI measurement—raw RSSI, RSSI filtered RSSI, and fusion filtered RSSI values. Figure 13 shows that after the distance of 7.5 m in fully indoor environment the fusion filter smoothed well the raw RSSI than RSSI filter. In case of half-indoor environment in the distance from 3 m to 7.3 m fusion filter's performance is better than the existing RSSI filter. In open-spaced indoor environment fusion filter gives better performance from 0 m to 5.8 m (Figure 15).

After applying the filter, the program analyses the behavior of the filtered signal strengths over distance.

4.3. BOTH Filter. Our last proposed filtering scheme is BOTH filter. For this filtering process, we used the same condition as fusion filter but changed the filtering method. In this experiment, if the difference of present RSSI and previous RSSI values is smaller than RSSI threshold value or if the LQI value is smaller than LQI threshold value then

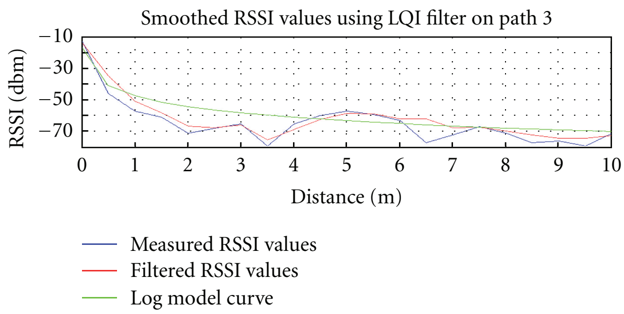


FIGURE 12: Smoothed RSSI values using LQI filter for path 3.

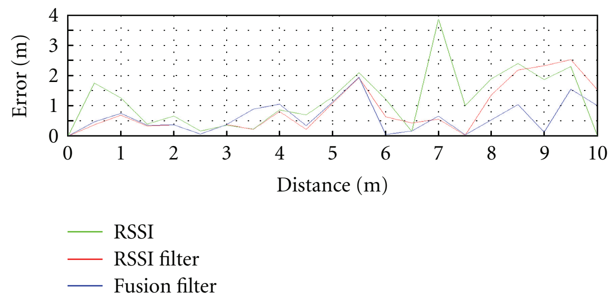


FIGURE 13: Distance error measurement of RSSI filter and fusion filter for path 1.

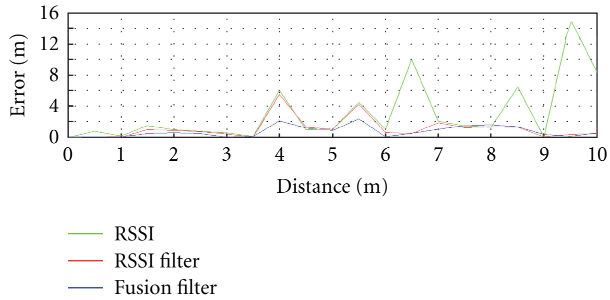


FIGURE 14: Distance error measurement of RSSI filter and fusion filter for path 2.

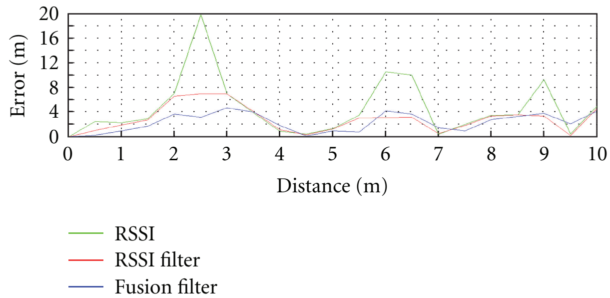


FIGURE 15: Distance error measurement of RSSI filter and fusion filter for path 3.

firstly the measured RSSI signal will be filtered by LQI filter. Secondly we used the RSSI filter to smooth the filtered RSSI by using RSSI filter. The following algorithm has been used to filter both RSSI and LQI values:

$$
\begin{aligned}
&\text{RSSI} \longrightarrow [\text{LQI filter}] \longrightarrow [\text{RSSI filter}] \\
&\qquad \longrightarrow \text{smooth\_RSSI}_{t(\text{Both})} \\
&\text{smooth\_RSSI}_{(\text{BOTH})} = a * \text{RSSI}_t + (1 - a) * \text{RSSI}_{(t-1)} \\
&\text{if } (\text{abs}(\text{RSSI}_t - \text{RSSI}_{t-1}) < \text{RSSI}_{\text{Threshold}}) \\
&\qquad \mid (\text{LQI}_\text{t} \ < \ \text{LQI}_{\text{Threshold}}) \\
&\qquad \text{LQI Filter} \\
&\text{smooth\_RSSI}_1 = \text{LQI Filter} \\
&\qquad \text{RSSI Filter} \\
&\text{smooth\_RSSI}_2 = \text{RSSI Filter} \\
&\text{smooth\_RSSI}_{t(\text{BOTH})} \ = \text{smooth\_RSSI}_2.
\end{aligned}
\tag{17}
$$

After applying the filter, the program analyses the behavior of the filtered signal strengths over distance. These figures are showing the distance error measurement of RSSI filter and fusion filter for 3 different environments. Figure 16 shows that the BOTH filtering smoothed all the sudden peaks and shaded signals from 4 m to 10 m well then the proposed LQI filter and existing RSSI filter. In Figure 17 also BOTH filter gave better performance from 1m to 7.3 m than LQI filter and RSSI filter. From Figure 18 we can see that again BOTH filter performed better than the LQI and RSSI filter in whole measured distance.

Based on this analysis, this system decides that LQI values perform better than RSSI values over distance in a fully indoor environment. But on the other two environments, RSSI values perform better than LQI values. The best performance happened when the RSSI values were smoothed by BOTH filtering. Figures 16–18 provided the results. So, from the above analyses, we can determine that BOTH filters perform better than the existing RSSI filter [18], fusion filter [13], and our proposed LQI filter.

5. Experimental Performance

From the above findings, we can say that the proposed BOTH filtering algorithm could reduce more average error and maximum error distance than other existing algorithm [8, 13, 18, 19]. The average reduction of distance error deduced by using the proposed BOTH filtering is 56% and 68% average error and maximum error, respectively, than the existing RSSI filtering [18] and fusion filtering [13]. As a result, our proposed method can perform well over other existing algorithm.

Table 2 shows the experimental results of our proposed filtering algorithm. From adaptive adjustment we found that this process could reduce a high number of errors without filter measured signal. So, we can draw the conclusion that our new enhancement technique gives a significantly improved performance over other existing techniques.

Table 2: Error reduction comparison of RSSI filter, fusion filter, proposed LQI filter, and BOTH filter.

Path	Filter name	Av. error	Max. error	RSSI filter		LQI filter		Fusion filter		BOTH filter	
1	Nonfilter	1.2750	4.0577								
	RSSI filter	1.0911	3.1117								
	LQI filter	1.0962	2.7127								
	Fusion filter	1.2807	3.0742								
	BOTH filter	1.0959	2.6453								
2	Non filter	3.0142	15.183								
	RSSI filter	1.0943	5.4847								
	LQI filter	1.0040	3.4574	Av. RDC. of Av. Error 33%	Av. RDC. of Max. Error 50%	Av. RDC. of Av. Error 41%	Av. RDC. of Max. Error 49%	Av. RDC. of Av. Error 49%	Av. RDC. of Max. Error 62%	Av. RDC. of Av. Error 56%	Av. RDC. of Max. Error 68%
	Fusion filter	0.8408	3.3281								
	BOTH filter	0.6495	1.8214								
3	Non filter	5.3016	21.183								
	RSSI filter	2.3972	7.2861								
	LQI filter	3.0213	13.609								
	Fusion filter	1.3088	3.1259								
	BOTH filter	1.3174	3.8351								

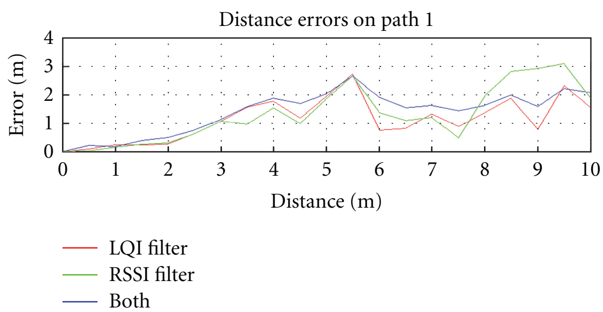


Figure 16: Smoothed RSSI values using LQI, RSSI, and BOTH filter for path 1.

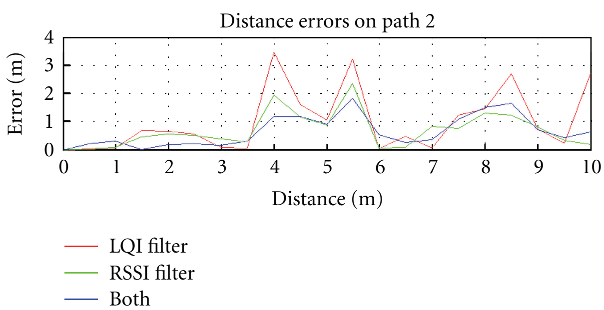


Figure 17: Smoothed RSSI values using LQI, RSSI and BOTH filter for path 2.

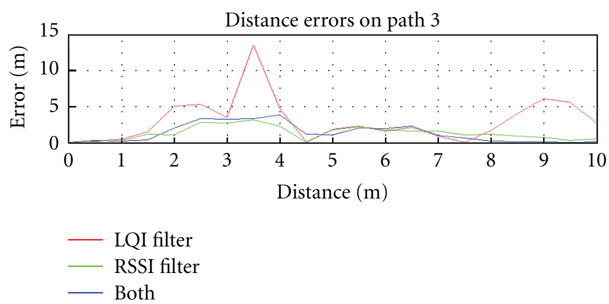


Figure 18: Smoothed RSSI values using LQI, RSSI, and BOTH filter for path 3.

6. Conclusion

The technology is developing faster and faster; the expansion of the internet makes people connected. There have been many forms of connections in cyberspace, that is, wired connection, wireless network, structured network, Ad hoc network, and so forth. Life will be rather different without any form of such communication. Certainly the security will be a great concern in such beneficial technology. The security measures to provide confidentiality and integrity have been taken into account in the design of such technology. This paper investigates the use of RF location systems for indoor domestic applications. Based on the assumption, low cost and minimal infrastructure are important for consumers; the concept of RF location system for Integrated Indoor Location Using RSSI and LQI provided by ZigBee module is introduced.

This paper addresses the problem of tracking an object and discusses how to overcome the problems in the existing methods calculating the distance in indoor environment. In this paper, a new mathematical method is presented for reducing the error in the location identification due to interference within the infrastructure-based sensor network. The proposed mathematical method calculates the distance using LQI and RSSI predicted based on the previously measured values. The calculated distance corrects the error induced by interference. The experimental results show that the proposed mathematical method can reduce the average error around 25%, and it is always better than the other existing interference avoidance algorithms. This technique has been found to work well in instances modeled on real world usage and thereby minimizing the effect of the error and hope that the findings of this paper will be helpful for design and implementation of object tracking system in indoor environment.

Acknowledgment

This paper was supported by the Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education, Science and Technology (2010-0024532).

References

[1] A. Krohn, M. Beigl, M. Hazas, H. W. Gellersen, and A. Schmidt, "Using fine-grained infrared positioning to support the surfacebased activities of mobile users," in *Proceedings of the 5th International Workshop on Smart Appliances and Wearable Computing (IWSAWC '05)*, Washington, DC, USA, 2005.

[2] N. Priyantha, A. Charraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the ACM Mobicomm*, pp. 32–43, New York, NY, USA, 2000.

[3] Y. Fukuju, M. Minami, H. Morikawa, and T. Aoyama, "Dolphin: an autonomous indoor positioning system in ubiquitous computing environment," in *Proceedings of the 1st IEEE Workshop on Software Technologies for Future Embedded Systems (WSTFES '03)*, pp. 53–56, May 2003.

[4] http://www.ubisense.net/.

[5] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.

[6] P. Bahl and V. Padmanabhan, "Radar: an in-building RF-based user location and tracking system," in *Proceedings of the IEEE Infocomm*, vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.

[7] A. Fink and H. Beikirch, "RSSI-based indoor localization using antenna diversity and plausibility filter," in *Proceedings of the 6th Workshop on Positioning, Navigation and Communication (WPNC '09)*, pp. 159–165, Hannover, Germany, March 2009.

[8] J. Jung and H. Myung, "Indoor localization using particle filter and map-based NLOS ranging model," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5185–5190, Shanghai, China, May 2011.

[9] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 517–526, October 2004.

[10] J. E. Berg, R. Bownds, and F. Lotse, "Path loss and fading models for microcells at 900 MHz," in *Proceedings of the IEEE 42nd Vehicular Technology Conference*, pp. 666–671, Denver, Colo, USA, May 1992.

[11] V. Seshadri, G. V. Zaruba, and M. Huber, "A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication," in *Proceedings of the IEEE PERCOM*, pp. 75–84, March 2005.

[12] R. Singh, L. Macchi, C. Regazzoni, and K. Plataniotis, "A statistical modelling based location determination method using fusion in WLAN," in *Proceedings of the International Workshop on Wireless Ad-Hoc Networks*, 2005.

[13] S. J. Halder, T. Y. Choi, J. H. Park, S. H. Kang, S. W. Park, and J. G. Park, "Enhanced ranging using adaptive filter of ZIGBEE RSSI and LQI measurement," in *Proceedings of the 10th International Conference on Information Integration and Web-Based Applications & Services (iiWAS '08)*, 2008.

[14] "CC2430 datasheet," http://www.chipcon.com/.

[15] "Wireless medium access control (MAC) and physical layer (PHY) specification for low-rate wireless personal area networks (LR-WPANs)," IEEE 802.15.4 Standard, 2003.

[16] S. Ko, J. Choi, and B. Kim, "Indoor mobile localization system and stabilization of localization performance using pre-filtering," *International Journal of Contro, Automation, and System*, vol. 6, no. 2, pp. 204–213, 2008.

[17] A. Dhital, P. Closas, and C. Fernandez, "Bayesian filters for indoor localization using wireless sensor network," in *Proceedings of the 5th ESA Workshop on Satellite Navigation Technologies (NAVITEC '10)*, pp. 1–7, Noordwijk, The Netherlands, December 2010.

[18] T. Y. Choi, *A study on in-door positioning method using RSSI value in IEEE 802.15.4 WPAN [M.S. thesis]*, School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu, Republic of Korea, 2007.

[19] S. J. Halder, T. Y. Choi, J. H. Park, S. H. Kang, S. J. Yun, and J. G. Park, "On-line ranging for mobile objects using ZIGBEE RSSI measurement," in *Proceedings of the 3rd International Conference on Pervasive Computing and Applications (ICPCA '08)*, vol. 2, pp. 662–666, Alexandria, Egypt, October 2008.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 107041, 11 pages
doi:10.1155/2012/107041

Research Article

A DHT-Based Discovery Service for the Internet of Things

Federica Paganelli and David Parlanti

CNIT, Research Unit at the University of Firenze, Via S. Marta 3, 50139 Firenze, Italy

Correspondence should be addressed to Federica Paganelli, federica.paganelli@unifi.it

Received 31 March 2012; Accepted 17 September 2012

Academic Editor: Jae-Ho Choi

Current trends towards the Future Internet are envisaging the conception of novel services endowed with context-aware and autonomic capabilities to improve end users' quality of life. The Internet of Things paradigm is expected to contribute towards this ambitious vision by proposing models and mechanisms enabling the creation of networks of "smart things" on a large scale. It is widely recognized that efficient mechanisms for discovering available resources and capabilities are required to realize such vision. The contribution of this work consists in a novel discovery service for the Internet of Things. The proposed solution adopts a peer-to-peer approach for guaranteeing scalability, robustness, and easy maintenance of the overall system. While most existing peer-to-peer discovery services proposed for the IoT support solely exact match queries on a single attribute (i.e., the object identifier), our solution can handle multiattribute and range queries. We defined a layered approach by distinguishing three main aspects: multiattribute indexing, range query support, peer-to-peer routing. We chose to adopt an over-DHT indexing scheme to guarantee ease of design and implementation principles. We report on the implementation of a Proof of Concept in a dangerous goods monitoring scenario, and, finally, we discuss test results for structural properties and query performance evaluation.

1. Introduction

The research roadmap towards the Future Internet is envisaging novel services endowed with context-aware and autonomic capabilities to support end users in daily living activities (e.g., work, leisure time, travel). In such a perspective, the technological landscape is expected to be populated by a wide range of functional capabilities offered by heterogeneous types of devices (PCs, mobile phones, household appliances, smart textiles, etc.). Several research fields are expected to contribute towards this ambitious vision, including the Internet of Things, the Internet of Services, and Cloud Computing.

The "Internet of Things" paradigm aims at providing models and mechanisms enabling the creation of networks of "smart things" on a large scale by means of RFID, wireless sensor and actuator networks, and embedded devices distributed in the physical environment [1]. This paradigm will open up the possibility to create novel value-added services by dynamically assembling different types of capabilities (sensing, communication, information processing, and actuation on physical resources, just to mention a few examples). Nonetheless, it is also well-known that the sustainable and efficient realization of IoT solutions requires the conception and development of dynamic adaptation and autonomic capabilities. Hence, the availability of mechanisms for discovering available resources and capabilities is of primary importance in the realization of the above-mentioned vision.

In the Internet of Things, "*data of real world objects and events will be available globally and in vast amounts. These data will be stored in widely distributed, heterogeneous information systems, and will also be in high demand by business and end user applications.*" [2]. Discovery mechanisms are thus required to enable a client application to obtain the location and addressing information needed to access these information repositories.

The contribution of this work consists in a distributed Discovery Service for Internet of Things scenarios. The proposed solution adopts a peer-to-peer approach for guaranteeing scalability, robustness, and maintainability of the overall system. While most peer-to-peer discovery services recently proposed for the IoT support exact match queries on a single attribute (i.e., the object identifier) [2], our solution can handle also multiattribute and range queries.

Indeed, while a one-attribute exact match query works well when a client application already knows the identifiers of the target objects (e.g., by means of readers that detect objects tagged with an RFID), the capability of handling multiattribute range query allows client applications to discover information and functional capabilities of objects that have not been detected yet and are not necessarily in physical proximity. The proposed solution is based on a layered architectural design that distinguishes three main aspects: multiattribute indexing, range query support, peer-to-peer routing.

The paper is organized as follows: in Section 2 we discuss functional requirements for discovery services in the IoT. Section 3 surveys related work. In Section 4 we describe the proposed distributed discovery service based on a P2P overlay. In Section 5 we present a Proof of Concept implementation in a reference application scenario and show testing results obtained by means of computational simulations. In Section 6 we draw the conclusions by discussing achieved results and future research directions.

2. Discovery Services in IoT Scenarios

The Internet of Things (IoTs) paradigm implies the perspective of objects and devices endowed with computational, sensing, and communication capabilities and capable of producing and disseminating a large amount of events. IoT applications should adapt their behaviour to changes in an extremely dynamic environment: users enter and exit from "smart environments" (e.g., smart homes [3–5], smart hospitals [6–8], smart offices [9, 10], and tourism locations [11, 12]), objects change their position according to specific application purposes (e.g., mobility of persons [13, 14], multimodal transport of goods [15], and maritime surveillance [16]). Discovery mechanisms are thus required in order to obtain up-to-date information about functional capabilities offered by devices distributed in the environment or information repositories storing information about a given object.

As argued by Atzori et al. [17], "key components of the IoT will be RFID systems." Radio frequency identification technologies (RFID) are typically composed by two types of hardware components: RFID tags and readers [18]. Tags store a unique identifier and, optionally, additional information. They can be applied to objects, animals, or persons (e.g., goods loaded on pallets or containers, smart bracelets worn by patients in hospitals [19]) to gather information about their status or surrounding environment (e.g., measurements gathered from sensors monitoring the status of goods or biomedical parameters of a target patient) and infer knowledge about the context (e.g., alarm and critical events) [20]. RFID and sensing technologies can thus be exploited in the development of context-aware applications in a wide range of application domains: logistic, e-health, security, smart cities [17].

As envisaged in a study promoted by the European Commission [21], it is foreseeable that any "thing" will have at least one unique way of identification (via a "Unique Identifier" or indirectly by some "Virtual Identifier" techniques). Endowed with addressing and communication capabilities, these things will be capable of connecting each other and exchange information. Information and services about objects will be fragmented and handled across many entities (ranging from the creator/owner of the object to the entities that have interacted with it at some stage in its lifecycle). Available information could be provided either at the level of the single object instance (e.g., a goods item) or a group or class of objects (e.g., items of dangerous goods that are transported by land, air, sea, or a combination of these).

Although several low-level requirements could emerge in the manifold application areas of IoT, we elicited the following high-level functional requirements for a discovery service in IoT scenarios [2].

2.1. Flexible Identification Scheme. The discovery mechanism should be transparent with respect to the adopted identification scheme. For instance, in RFID applications for logistics, the Electronic Product Code (EPC) [22] is a widely adopted identification scheme. Other available identification schemes include URIs, IPv6 addresses, Universal Product Code, just to mention a few examples.

2.2. Multiattribute Query. The discovery mechanism should be capable of handling a query for an exact match of a given identifier as well as queries possibly containing other qualifying attributes (e.g., location and category).

2.3. Range Query. In addition to exact match queries, the system should support queries specifying lower and upper boundaries on a single or multiple attributes.

2.4. Multiple Publishers. Depending on the application purposes, several entities may be called to produce and publish information about a given object, besides the object's owner.

2.5. Management APIs. Authorized entities should be able to add, update and delete information associated with a given object.

3. Related Work

RFID systems are considered key components of the IoT [17]. A significant standardization effort has been performed by the EPCglobal consortium to establish principles and guidelines for supporting the use of RFID in trading and enterprise contexts. More specifically, the EPCglobal Network is a set of emerging standard specifications for a global RFID data sharing infrastructure built around the Electronic Product Code (EPC), an unambiguous numbering scheme for the designation of physical goods [22]. It aims at facilitating the handling, storage, and retrieval of information related to EPC-identified items.

The EPC is a universal identifier used for physical objects. It can take the form of a Uniform Resource Identifier (URI), thus enabling information systems to refer to physical objects. The EPC code is typically stored on an RFID attached to the referred object. Main components of the EPC Global

Architecture include the RFID Tags, the Readers, the EPC Middleware, the EPC Information Services (EPCISs), the Object Naming Service (ONS), and Discovery Services. The specifications define how Readers interrogate an RFID tag. The Middleware filters and processes data that are gathered by Reader components. Data are then stored in EPCIS repositories and made available to external clients via the EPCIS Query Interface. The ONS offers a name resolution service that translates an EPC code into the URLs pointing to the EPCIS repositories storing data about that EPC. More specifically, static and dynamic data about physical objects are stored in databases that can be handled by different actors (e.g., manufacturers, logistic providers, retailers, or third parties) and can be accessed via the EPCIS standard interface [23].

The EPC Object Name Service (ONS) is the service that provides clients with the EPCIS URL for a given EPC. The ONS is based on the Internet Domain Name System, which is characterized by a hierarchical architecture [24]. The ONS points to the manufacturer's EPCIS resources. Moreover, discovery services enable discovery of third parties' EPCIS repositories. The EPCIS Discovery Service is the lookup service providing clients with the URIs of EPCIS repositories storing information about a given EPC. The EPCIS Discovery specifications have not been published yet at the time of writing [25]. Several research contributions have thus attempted to fill this gap by proposing original solutions for discovery services.

The BRIDGE Project, funded by the European Commission, had the objective of investigating several issues related to the implementation and adoption of RFIDs in Europe. In the framework of the BRIDGE project, a prototype discovery service was implemented based on LDAP directories specifications [26]. The authors of [27] proposed an implementation of an EPC discovery service based on the IETF specifications of the Extensible Supply-chain Discovery Service ESDS [28].

More recently, some works investigated the use of peer-to-peer (P2P) systems to implement scalable and robust distributed discovery services. Schoenemann et al. [29] proposed a P2P-based architecture for enabling the information exchange among participants of a supply chain. Analogously, Shrestha et al. [30] proposed a peer-to-peer network, where each participant of the supply chain runs a node of the network. These nodes form a structured P2P overlay network with each node having a partial view of the other nodes. Manzanares-Lopez et al. [31] proposed a distributed discovery service for the EPCglobal network based on a P2P architecture, which offers item-level track and trace capabilities along the whole supply chain, also when items are not directly visible, since they are loaded within different storage systems (i.e., packages, boxes, containers, etc.). These peer-to-peer approaches typically adopt Distributed Hash Table (DHT) techniques. Distributed Hash Tables are distributed data structures where the information objects are placed deterministically, at the peer whose identifier corresponds to the information object's unique key according to a given distance metric [32].

In a DHT, each node is identified by means of a key (node-key), usually the MAC or IP address of the node. Analogously, content items handled by the network are identified by a key (content key). Both types of keys are mapped to an identifier of a given bit length by applying a hash function (e.g., SHA-1 or MD5). DHT overlay networks implementations differ in how nodes and contents are associated and how routing to the node responsible for a given identifier is defined. Well-known routing algorithms are Chord [33], Pastry [34], Tapestry [35], and Kademlia [36].

The study carried out by Evdokimov et al. [2] discusses some of the above-mentioned works [23, 24, 26, 29] by comparing strengths and weaknesses of these works and highlighting how P2P approaches better fulfil fault tolerance and communication scalability requirements. As discussed in that study, DHT peer-to-peer networks exhibit properties of scalability, efficiency, robustness, and load-balancing thanks to the adoption of consistent hashing. Typically, in a DHT of N nodes a lookup operation requires O $(\log N)$ hops. These networks show properties of self-organization and self-healing, since they are capable of handling joining and leaving events of participating members. For this reason, they also guarantee resilience to node failures and network malfunctioning. Load balancing is achieved through uniform hashing.

As highlighted in the state of the art analysis made by Evdokimov et al. [2], the above-mentioned works support queries providing an object identifier (typically the EPC code) as input, but they do not support more flexible query schemes based on object attributes, though these types of information queries could become very important in the future IoT.

4. A Discovery Service for the IoT Based on a Peer-to-Peer Network

This section describes our discovery service architecture based on a peer-to-peer overlay network. In particular, our approach aims to cope with the following functional requirements: flexible identification scheme, multiattribute query, range query, multiple publishers, management APIs. As discussed in the Related Work Section, most existing works do not support multiattribute and range queries. These contributions are able to solve an exact match query and thus they assume that the client knows the identifier of the target object. This assumption can be easily satisfied if the client application can acquire the identifier from an RFID reader that is close to the RFID-tagged object. However, such a physical proximity constraint would inevitably limit the scope of IoT applications. For instance, a good monitoring application might include the following desired features: (a) displaying on the map the current and past positions of a goods item with identifier I during a multimodal transport route, (b) displaying on the map the current and past positions of the items travelling from location X to location Y. In this example, feature (b) might be easily realized by relying on a discovery service capable of handling complex queries, in addition to exact match queries.

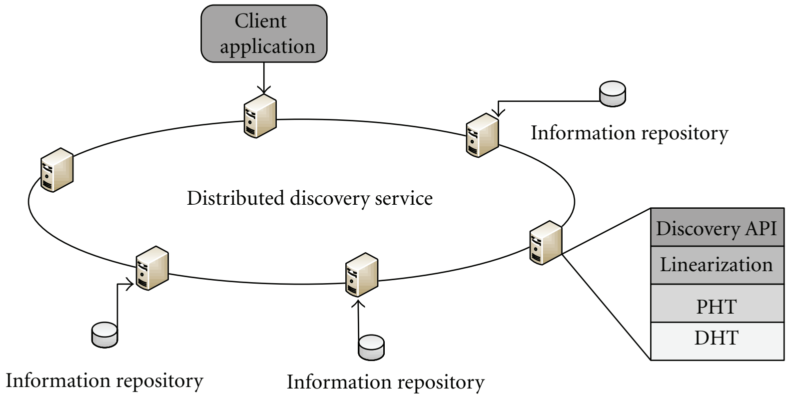


Figure 1: Functional view of the proposed Distributed Discovery Service.

Our work aims to address these limitations of related work by proposing a distributed discovery service capable of handling multiattribute and range queries. Hereafter, we describe the design of our DHT-based discovery service and our choice to adopt a layered functional architecture to promote modularity, ease of design and implementation, and maintainability of the system.

4.1. Existing DHT Implementations for Handling Complex Queries. Most widely adopted DHT implementations, such as Chord [33] and Kademlia [36], support one-attribute exact match queries. Several solutions have been proposed in order to handle range queries. The approach proposed in [21] is based on the adoption of the Prefix Hash Tree (PHT), a distributed data structure that can be built on top of a DHT implementation. The PHT overlay is based on a trie-based structure. A trie is a special tree in which each node represents a prefix of the target data domain. Interestingly, PHT relies merely on the DHT lookup() operation, and it is hence agnostic to the underlying DHT algorithm and implementation. The PHT solution presented in [37] supports only single-attribute range queries.

Mercury [38] supports multiattribute range queries by handling multiple simple overlays, one for each attribute, mapped onto a set of physical nodes. MAAN [39] extends Chord with locality preserving hashing and a recursive multidimensional query resolution mechanism. MAAN relies on a locality-preserving hashing function for each attribute, which has to be constructed using the attribute's values distribution (to be known in advance).

Squid [40] is a peer-to-peer information discovery system implementing a DHT-based structured keyword search. Each data element is associated with a sequence of keywords, the keywords form a multidimensional keyword space where data elements are points in the space and the keywords are the coordinates. In Squid, Space Filling Curves are used to map this multidimensional keyword space to a one-dimensional index space. Space Filling Curves can be defined as a continuous mapping from a d-dimensional space to a one-dimensional space, that is, $f : N^d \rightarrow N$. Examples of SFCs are the Morton curve (z-curve), the Gray code curve, and the Hilbert curve. Depending on the adopted mapping rule, SFCs show different locality-preserving properties. SFCs are locality preserving in that points that are close in the one-dimensional space are mapped from close points in the d-dimensional space. In [41] a SFC-based technique is applied on an indexing scheme built on top of a generic DHT implementation to resolve multiattribute range queries.

4.2. Layered Architecture. We chose to use the Prefix Hash Tree (PHT) distributed data structure. Since a PHT data structure can be built on top of a generic DHT implementation, major advantages of this approach are the promotion of a layered design, and thus, modularity, ease of design, implementation, and maintenance.

Our design approach distinguishes the following layers: (a) an SFC linearization technique for mapping a multidimensional domain into a one-dimensional one, (b) a PHT search structure leveraging on the generic DHT get/put interface, (c) a DHT implementation based on the Kademlia algorithm [20]. By exploiting the distributed data management capabilities offered by this peer-to-peer overlay network, application-specific APIs for search and management of discovery-related information can be built.

Figure 1 shows the high-level architecture of the proposed system: each peer exposes a set of discovery APIs that can be invoked by client applications to search for some objects. The Discovery services returns the URIs of the repositories that handle information and services about those objects.

Below we describe our choices for the design of the three layers (a), (b), and (c), while Discovery APIs for a reference application scenario are described in Section 5.

4.2.1. Linearization of the Multiattribute Domain. We suppose that objects can be characterized by d attributes, including the object identifier (e.g., EPC, URI identification schemes). Consequently, our target data domain is a d-dimensional space. We applied a linearization technique based on Space Filling Curves to map a d-dimensional data domain onto a 1-dimensional data domain. The derived 1-dimensional data domain can thus be easily indexed in the PHT structure, as described below.

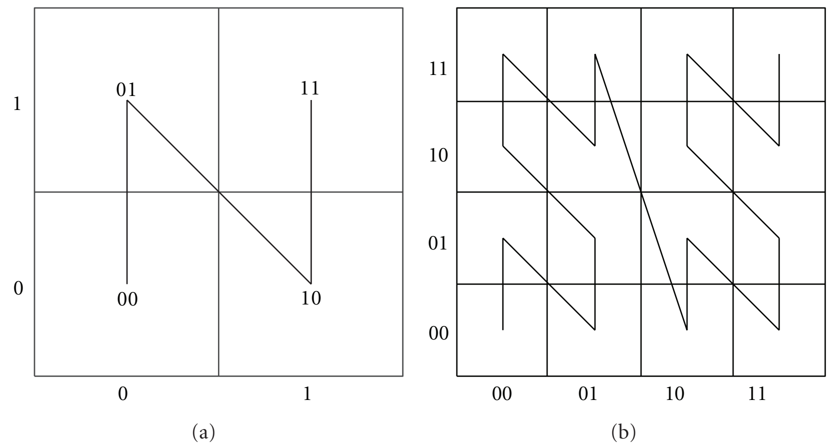


FIGURE 2: z-order curve approximations: (a) first-order z-curve; (b) second-order z-curve.

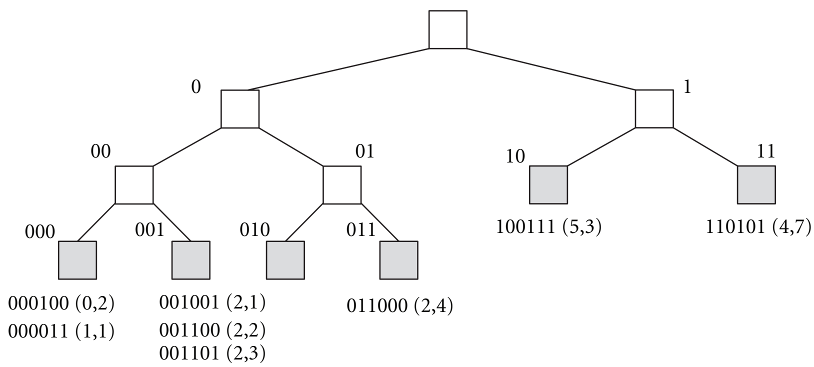


FIGURE 3: Example of a PHT for a two-dimensional data domain.

Among the existing types of SFC, we chose to adopt the Morton curve, known also as z-curve, since it is computationally simpler to generate than other known SFCs. A z-order-derived key is assembled by cyclically taking a bit from each coordinate of a point in d-dimensional space and appending it to those taken previously ("bit interleaving") [42]. For instance, the 2-dimensional point $P\ (x_1x_2x_3x_4\cdots x_d, y_1y_2y_3y_4\cdots y_d)$ is mapped onto the derived key $x_1y_1x_2y_2x_3y_3x_4y_4\cdots x_dy_d$.

A Space Filling Curve is constructed recursively and can be characterized by two parameters: the order of multidimensionality d and the order of approximation k. The first approximation (first-order curve) is obtained by partitioning a d-dimensional cube into n^d subcubes and connecting the subcubes to form a continuous curve. The kth approximation is obtained by connecting n^{kd} subcubes. Figure 2 shows an example of a SFC built in a 2-dimensional space over the binary domain $\{0, 1\}$: Figure 2(a) shows the first-order approximation and Figure 2(b) shows the second-order approximation.

4.2.2. PHT Data Structure. We assume that the data set to be indexed is some number N of D-bit binary keys. A PHT data structure is a binary trie, in which each node of the trie is labelled with a prefix. The prefix is defined recursively: a node with label l has either zero or two children, and the right and left child nodes are labelled with $l0$ and $l1$, respectively. A key K is stored at the leaf node whose label is a prefix of K. Each leaf node can store at most B keys. Consequently, the shape of the PHT depends on the distribution of keys: the trie depth is greater in regions of the domain that are densely populated and lower in sparsely populated domain regions. Figure 3 shows an example of a 2-dimensional data set indexed in a trie-based structure: first, 2-dimensional data objects are mapped into one-dimensional binary keys by means of the z-order curve linearization technique; then, the data objects are stored in the PHT leaf nodes whose label is a prefix of their one-dimensional binary key.

The PHT structure is an indexing data structure built on top of the DHT overlay network. As described in [21], the logical PHT structure is distributed across the nodes in a DHT. PHT prefixes are hashed and assigned to the proper DHT node, that is, a PHT vertex with label l is assigned to the node whose identifier is closest to HASH(l). The PHT can be built by relying on two basic DHT operations: put(key, value) and get(key) primitives. It does not require knowledge of the DHT routing algorithm and implementation details. Consequently, it can be built on top of any DHT implementation.

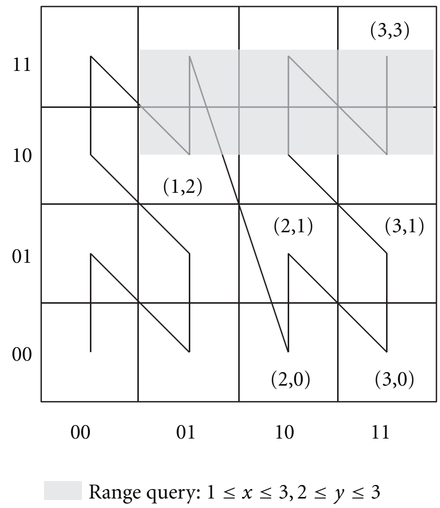


Figure 4: Results for a range query over a one-dimensional z-curve.

PHT Lookup Operation. Given a key K of D bits, a PHT lookup operation shall return the unique leaf node whose label is a prefix of K. According to the algorithm proposed in [37], we adopted a binary search approach. The first step is to determine the prefix from which to start the search, so we consider the first $D/2$ bits of key K. Then, the get() operation is invoked on the DHT overlay with the $D/2$ key prefix provided as input parameter. The result returned by the DHT can be of three types: null, internal node, or leaf node. If the retrieved node is a leaf node, the search algorithm returns all the keys, along with the desired key, stored in the node; if the node does not exist (null), the prefix length is reduced. If the prefix is associated to an internal node, the search algorithm tries longer prefixes, repeating this step recursively until a leaf node is found. This kind of search has been preferred to a linear approach since it allows to considerably reduce the workload on the root and it requires $\log D$ DHT gets, thus making the lookup operation efficient [37].

PHT Insert/Delete Operation. The insertion or deletion of a key K in the PHT relies on the lookup operation to find the proper leaf node. In case of insertion, if the constraint on the number of maximum allowed children is violated, the node has to be split into two nodes, and the keys stored in the original node are distributed across the two novel leaf nodes.

PHT Range Query. In a one-dimensional data domain, a range query can be expressed as follows: given two keys L and H, where $L < H$, a range query shall return all the keys K that satisfy the condition $L \leq K \leq H$.

The PHT search algorithm first defines the smallest prefix that covers the whole specified range. If this prefix is the label of an internal node, then the query is forwarded recursively to the child nodes whose prefix falls within the specified range, until leaf nodes are reached [37].

An analogous approach is adopted for a range query in a d-dimensional domain. Minimal and maximal values for each coordinates are linearized to obtain minimum and maximum derived keys. Then, the maximum common prefix of these derived keys can be obtained. The maximum common prefix thus allows locating the node the search is started from. The search is forwarded recursively to child nodes, selecting those nodes whose prefix falls within the monodimensional range delimited by the minimum and maximum derived keys, until leaf nodes are found.

The above-mentioned search algorithm can return leaf nodes that contain keys whose delinearized d-dimensional value does not fall within the original range query. This fact depends on the locality-preserving property of the adopted SFC curve. Points that are close on a linearized monodimensional space can be distant in the originating d-dimensional one. As shown in Figure 4, a range query in a 2-dimensional domain ($1 \leq x \leq 3$, $1 \leq y \leq 3$) can be mapped into a range query on a 1-dimensional binary domain ($0111 \leq z \leq 1111$) by applying the z-order curve linearization technique, but some results of the 1-dimensional query fall outside the original 2-dimensional query.

As searches of the left and right subtrees can be performed in parallel, the complexity of the algorithm is linearly proportional to the depth of the trie region that has been analysed.

If the query range is increased, the length of the maximal common prefix where the search starts from decreases. Consequently, the cost of the search increases since the trie region to be analysed increases. A typical example is when a range query is specified for a subset of attributes (e.g., x attributes), while for the remaining $d - x$ attributes any value is allowed, or a wild card query is performed. In order to reduce the cost of search, we introduced a mechanism that allows starting the search in parallel branches at deeper levels in the trie, by exploiting a simple pattern matching technique. First, we defined a “relaxed common prefix”. Given the minimum and maximum derived keys, L ($l_1 l_2 l_3 \cdots l_k$) and H ($h_1 h_2 h_3 \cdots h_k$), respectively, the relaxed maximal common prefix P ($p_1 p_2 p_3 \cdots p_k$) is built according to Pseudocode 1.

```
Input:
minimum derived key L (l1 l2 l3 · · · lk)
maximum derived key H (h1 h2 h3 · · · hk),
Output:
relaxed maximal common prefix P (p1 p2 p3 . . . pm) with m <= k
counter i ← 0;
counter j ← 0;
WHILE (hi ≥ li AND i < n) DO
{
IF hi = li THEN pj = hi;
IF hi = 1 AND li = 0 THEN pj = . (where. is the wildcard bit);
i + 1;
j + 1;
}
```

Pseudocode 1

For instance, in a bidimensional data domain, the range query $2 \leq x \leq 4$, $2 \leq y \leq 4$ originates the minimum and maximum derived keys 001100 and 110000, respectively. While the maximum common prefix would be null, the "relaxed common prefix" is "..", where the symbol "." represents a wildcard bit (i.e., it matches any value {0, 1}). The search can thus be performed in parallel by starting from the prefixes matching the relaxed common prefix.

4.2.3. DHT. As mentioned above, the PHT indexing scheme is independent from the underlying DHT implementation, since a PHT relies on the basic put() and get() primitives offered by any DHT. The DHT that has been used for this work has been design and developed from scratch based upon the Kademlia specifications [20]. In addition, our DHT implementation supports data replication and versioning. Replicas of each data item are maintained in N nodes of the DHT network, where N can be configured. Replication is needed to improve availability and fault tolerance, while it introduces threats to the consistency of stored data. Our system implements an eventual consistency approach by handling multiple versions of a given data item. Conflicts arising from the presence of multiple versions of a data item in the system at the same time are handled and resolved by using vector clocks [43]. A vector clock is a list of tuples (node address, counter), and each version of an object has a vector clock assigned. Vector clocks can be analysed to infer causality between two events (i.e., different versions of the same object) and thus to decide if two versions of the same object are on parallel versioning branches (i.e., they are conflicting resources). When a versioning conflict is detected, the two resources are reconciled. In the current implementation, a basic reconciliation mechanism is implemented by merging the conflicting versions.

5. Experimentation

We performed a set of experimentation activities in a reference scenario for goods tracing and tracking in a multimodal transport chain.

In this reference scenario, several types of actors are involved, as shown in Figure 5: multimodal transport operators, road transport operators, shipping companies, intermodal terminal operators, and customers (the sender and addressee). These actors typically interact with tracking and tracing services to query for or to insert new information about the status and location of monitored good items. Moreover, also institutional actors (e.g., Port Authorities, Customs systems, etc.) and third-party actors (e.g., banks, insurance companies) can query for information about goods on transit [44].

In this scenario, we defined a set of attributes, in addition to the goods item identifier (i.e., the EPC number), that can usefully characterize the information acquired and stored during the steps of the transport route and that can be used to ease information retrieval tasks for different application purposes.

(i) departure latitude;

(ii) departure longitude;

(iii) arrival latitude;

(iv) arrival longitude;

(v) type of goods (e.g., dangerous goods classification codes).

Our solution does not mandate any specific Object Identifier schema. For this experimentation activity we chose the EPC-64 numbering scheme (an EPC code encoded as a sequence of 64 bits). Latitude and longitude coordinates can be encoded into a sequence of 40 bits each [45]. For the type of goods attribute, we adopted the UN classification schema for dangerous goods, consisting in a 4-digit number, encoded in 16 bits. As the linearization technique has to be applied to sequences of bits of equal length, the shorter sequences are padded with zero bits on the left, so that they have the same length as the longer sequence (i.e., the 64-bit EPC code). The resulting PHT key is thus 384-bit long.

The information record stored for each key is a list of tuples (URL, timestamp), where the URL refers to an information repository storing the information about the target

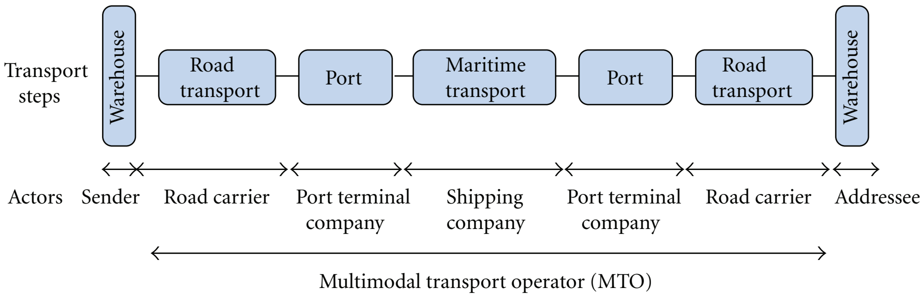


Figure 5: Actors involved in a multimodal transport chain.

object, and the timestamp marks the time when the record has been inserted in the system. As a matter of fact, during an object lifecycle, several information repositories handled by different actors could be associated to the object's identifier.

The API exposed to client applications can thus support the following operations.

(i) Inserting a new object: the client has to provide the values of the selected six attributes. These values are processed by the system to generate the corresponding PHT key. The system inserts the PHT key in the trie-based structure and the associated information (URL and timestamp) in the underlying DHT nodes.

(ii) Adding/deleting an information record associated to an object: the system uses the PHT key associated with the object to locate the information records stored in the DHT and update them by adding/deleting the given record.

(iii) Retrieving the information records for a given object: the system uses the PHT key associated with the provided query input parameters to locate the DHT nodes and retrieve stored information records.

(iv) Retrieving the information records for a set of objects that satisfy a range query over (a set of) attributes: the system exploits the trie-based structure to retrieve the PHT keys that are within the query range and to gather the information objects associated with the retrieved objects.

We performed a set of testing activities in order to analyse the structural properties and performance of the PHT. The object of our testing activities was the PHT overlay and not the underlying DHT implementation, therefore we adopted the testing methodology proposed in [41]. We used a DHT network of 20 nodes running on two physical hosts connected on a LAN environment. Since performance is measured in terms of DHT operations, we can abstract from the network characteristics.

To analyse the structural properties of the PHT indexing scheme, we conducted a set of experiments through computational simulations. We adopted two metrics: the average leaf depth of the trie and the average block utilization, which is calculated as the ratio of the number of elements stored in the leaf node to the value of block B, that is, the maximum number of keys that can be stored in a node [41]. We measured these properties on a data set of progressively increasing size (up to 20,000 keys) populated with randomly generated keys.

Figure 6 shows how the average depth of the leaf nodes varies with the block size. The average depth of the nodes decreases logarithmically with the increase of the block size. This is due to the fact that increasing values of B (block size) results in leaves containing more keys and thus a less deep trie structure. The average depth of leaf nodes increases with the data set size. Figure 6 shows the average depth measured for data sets of size 90 kB, 400 kB, 600 kB, and 1 MB. This behaviour is analogous to the one observed for other over-DHT indexing approaches [41, 46].

As shown in Figure 7, the block utilization exhibits a fluctuating behaviour as the block size increases. The bucket utilization shows how full the leaf nodes are with respect to the maximum allowed block size. Especially for smaller data sets (90 kB and 600 kB), the block utilization value fluctuates as the block size increases. As the data set size increases, the block utilization tends to increase with the block size. These results can be hardly compared with other approaches in the literature, since the behaviour of this structural property may vary with the type of the data set distribution [46].

We measured the performance of insert and lookup operations in the PHT by analysing the number of accesses to the underlying DHT for invoking get() and put() operations. We populated the PHT with 200,000 derived keys, and we measured the number of DHT operations that were performed at each level of the PHT trie for inserting a set of 1,000 uniformly distributed keys. As depicted in Figure 8, results show that higher levels in the trie are seldom accessed, thanks to the adopted binary search approach. Then, we performed a set of range query operations over a population of 200,000 keys stored in the PHT. For each iteration, we varied the range span of the queries (from wider ranges with a common prefix of zero length to an exact match query). Again, the results show that the workload in terms of accesses to the DHT seldom affects the root of the trie (Figure 9).

According to these testing results, our system behaves analogously to other over-DHT indexing approaches [41, 46]. Overlay-dependent indexing schemes can implement more efficient mechanisms for handling complex queries, since they can also modify the underlying routing mechanism, such as Mercury [38]. However, they are usually more

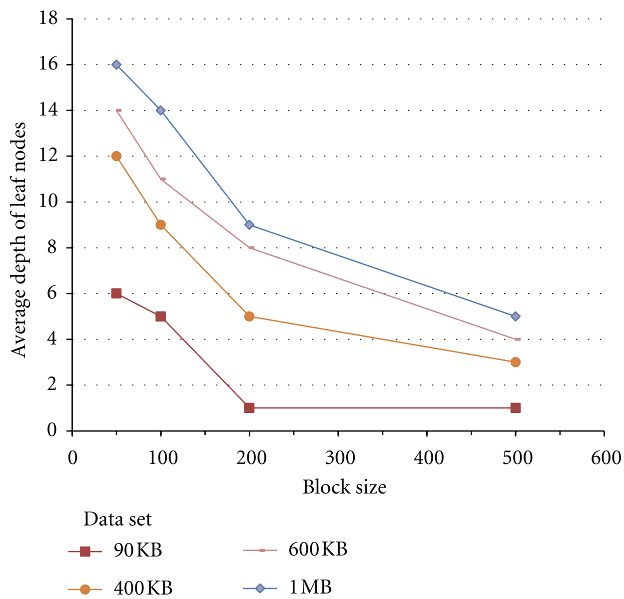


Figure 6: PHT structural properties: average depth of leaf nodes against block size for increasing data sets.

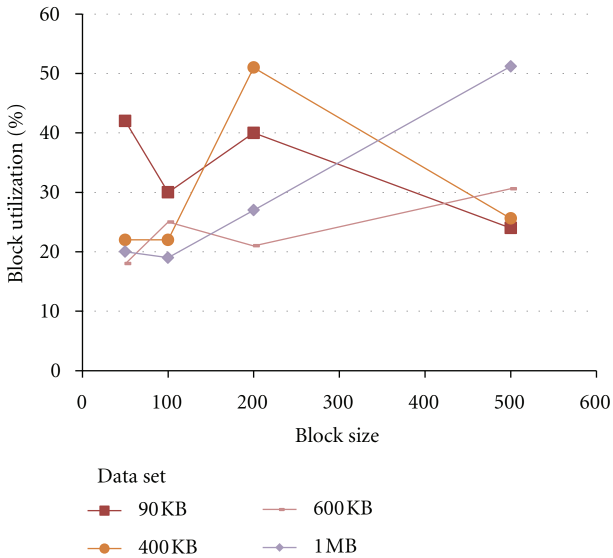


Figure 7: PHT structural properties: block utilization.

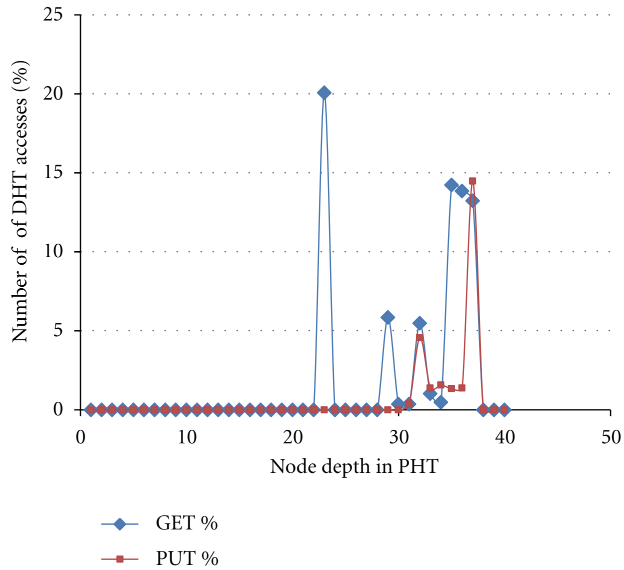


Figure 8: Number of accesses to the DHT against node depth in the PHT for the insert operation.

demanding in terms of complexity of design, development, and maintenance [46].

6. Conclusions

In this paper, we proposed a distributed Discovery Service based on a peer-to-peer overlay network for IoT scenarios. With respect to existing approaches implementing a DHT-based discovery service for the IoT and, in particular, for RFID-based scenarios, our original contribution consists in supporting more complex queries, that is, multiattribute and range queries.

Our design approach was based on the adoption of an over-DHT indexing scheme, thus easing the design of a layered functional architecture. More specifically, our discovery system design is made of the following layers: (a) an SFC linearization technique for mapping a multidimensional domain into a one-dimensional one, (b) a PHT search structure leveraging on a generic DHT get/put interface, (c) a DHT implementation based on the Kademlia algorithm. Although overlay-dependent indexing schemes can be more efficient in handling complex queries, they are usually more demanding in terms of complexity of design, development,

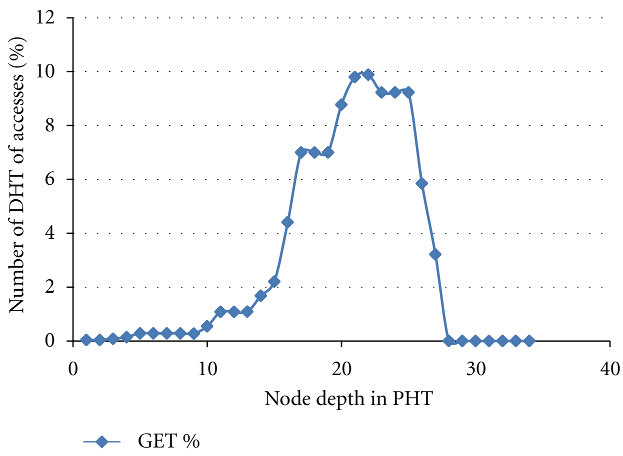


Figure 9: Number of accesses to the DHT against node depth in the PHT for the range query operation.

and maintenance [46]. Instead, our layered solution privileges ease of design and implementation.

We implemented a Proof of Concept in a reference application scenario for dangerous goods monitoring. We thus reported results achieved through experimentation activities regarding structural properties and query performance in an in-laboratory testing configuration. More extensive testing activities are planned in the near future. To this purpose, we are evaluating the possibility of exploiting the capabilities offered by PlanetLab, which is a large-scale distributed testbed [47].

Future research activities will also be devoted to carry out a case study on dangerous goods monitoring within a research project (SITMAR research project), funded by the Italian Ministry for Economic Development. To this purpose, we will also delve into security issues, ranging from secure communication to trust between interacting parties.

Acknowledgment

The authors thank Mr. Luca Capannesi from the University of Florence for his technical support.

References

[1] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," in *Proceedings of the 2nd International Internet of Things Conference (IoT '10)*, pp. 9–129, December 2010.

[2] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann, "Comparison of Discovery Service architectures for the Internet of Things," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '10)*, pp. 237–244, June 2010.

[3] G. D. Abowd, I. Bobick, I. Essa, E. Mynatt, and W. Rogers, "The aware home: developing technologies for successful aging," in *Proceedings of the American Association of Artificial Intelligence Conference*, 2002.

[4] F. Paganelli and D. Giuli, "An ontology-based system for context-aware and configurable services to support home-based continuous care," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 324–333, 2011.

[5] F. Paganelli and D. Giuli, "A context-aware service platform to support continuous care networks," in *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction (UAHCI '07)*, vol. 4555, part 2 of *Lecture Notes in Computer*, pp. 168–177, 2007.

[6] J. E. Bardram, "Applications of context-aware computing in hospital work—examples and design principles," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 1574–1579, March 2004.

[7] F. Paganelli, E. Spinicci, A. Mamelli, R. Bernazzani, and P. Barone, "ERMHAN: a multi-channel context-aware platform to support mobile caregivers in continuous care networks," in *Proceedings of the IEEE International Conference on Pervasive Services (ICPS '07)*, pp. 355–360, July 2007.

[8] F. Paganelli and D. Giuli, "An ontology-based context model for home health monitoring and alerting in chronic patient care networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and ApplicationsWorkshops/Symposia (AINAW '07)*, pp. 838–845, May 2007.

[9] F. Paganelli, M. C. Pettenati, and D. Giuli, "A metadata-based approach for unstructured document management in organizations," *Information Resources Management Journal*, vol. 19, no. 1, pp. 1–22, 2006.

[10] A. Kocurova, S. Oussena, P. Komisarczuk, and T. Clark, "Context-aware content-centric collaborative workflow management for mobile devices," in *Proceedings of the 2nd International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA '12)*, pp. 54–57, IARIA.

[11] A. García-Crespo, J. Chamizo, I. Rivera, M. Mencke, R. Colomo-Palacios, and J. M. Gómez-Berbís, "SPETA: social pervasive e-Tourism advisor," *Telematics and Informatics*, vol. 26, no. 3, pp. 306–315, 2009.

[12] F. Paganelli, G. Bianchi, and D. Giuli, "A context model for context-aware system design towards the ambient intelligence vision: experiences in the eTourism domain," in *Proceedings of the 9th ERCIM Workshop "User Interfaces For All", Special Theme: 'Universal Access in Ambient Intelligence Environments'*, Lecture Notes in Computer Science, Springer, Königswinter, Germany, September 2006.

[13] D. Giuli, F. Paganelli, S. Cuomo, and P. Cianchi, "A systemic and cooperative approach towards an integrated infomobility system at regional scale," in *Proceedings of the IEEE International Conference on ITS Telecommunications (ITST '11)*, pp. 547–553.

[14] J. Santa and A. F. Gómez-Skarmeta, "Sharing context-aware road and safety information," *IEEE Pervasive Computing*, vol. 8, no. 3, pp. 58–65, 2009.

[15] S. Turchi, L. Ciofi, F. Paganelli, F. Pirri, and D. Giuli, "Designing EPCIS through linked data and REST principles," in *Proceedings of the International Conference on Software, Telecommuniccations and Computer Networks (SoftCOM '12)*, Split, Croatia, September 2012.

[16] D. Parlanti, F. Paganelli, and D. Giuli, "A service-oriented approach for network-centric data integration and its application to maritime surveillance," *IEEE Systems Journal*, vol. 5, no. 2, pp. 164–175, 2011.

[17] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[18] K. Finkenzeller, *RFID Handbook*, Wiley, 2003.

[19] W. Yao, C. H. Chu, and Z. Li, "Leveraging complex event processing for smart hospitals using RFID," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 799–810, 2011.

[20] I. Zappia, F. Paganelli, and D. Parlanti, "A lightweight and extensible Complex Event Processing system for sense and respond applications," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10408–10419, 2012.

[21] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, *Vision and Challenges for Realising the Internet of Things*, Cerp-IoT Cluster of European Research Projects on the Internet of Things, European Commission, 2010.

[22] F. Thiesse, C. Floerkemeier, M. Harrison, F. Michahelles, and C. Roduner, "Technology, standards, and real-world deployments of the EPC network," *IEEE Internet Computing*, vol. 13, no. 2, pp. 36–43, 2009.

[23] B. Fabian and O. Günther, "Security challenges of the EPCglobal network," *Communications of the ACM*, vol. 52, no. 7, pp. 121–125, 2009.

[24] EPCglobal, Object Name Service (ONS) 1.0.1, Ratified Standard Specification with Approved, Fixed Errata, 2008.

[25] EPCGlobal, http://www.gs1.org/gsmp/kc/epcglobal.

[26] BRIDGE Project, "Working prototype of serial-level lookup service," 2008, http://www.bridge-project.eu/data/File/BRIDGE_WP02_Prototype_Serial_level_lookup_service.pdf.

[27] U. Barchetti, A. Bucciero, M. De Blasi, L. Mainetti, and L. Patrono, "Implementation and testing of an EPCglobal-aware discovery service for item-level traceability," in *Proceedings of the International Conference on Ultra Modern Telecommunications and Workshops (ICUMT '09)*, pp. 1–8, October 2009.

[28] M. Young, "Extensible supply-chain discovery service concepts (Draft 04)," Internet Draft, IETF, 2008.

[29] N. Schoenemann, K. Fischbach, and D. Schoder, "P2P architecture for ubiquitous supply chain systems," in *Proceedings of the 17th European Conference on Information Systems*, pp. 2255–2266, 2009.

[30] S. Shrestha, D. S. Kim, S. Lee, and J. S. Park, "A peer-to-peer RFID resolution framework for supply chain network," in *Proceedings of the 2nd International Conference on Future Networks (ICFN '10)*, pp. 318–322, January 2010.

[31] P. Manzanares-Lopez, J. P. Muoz-Gea, J. Malgosa-Sanahuja, and J. C. Sanchez-Aarnoutse, "An efficient distributed discovery service for EPCglobal network in nested package scenarios," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 925–937, 2011.

[32] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.

[33] I. Stoica, R. Morris, D. Liben-Nowell et al., "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[34] A. Rowstron and P. Druschel, "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware '01)*, pp. 329–335, Springer, London, UK, 2001.

[35] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: a resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, 2004.

[36] P. Maymounkov and D. Mazieres, "Kademlia: a peer-to-peer information system based on the XOR metric," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '01)*, pp. 53–65, Springer, London, UK, 2002.

[37] S. Ramabhadran, S. Ratnasamy, J. M. Hellerstein, and S. Shenker, "Brief announcement: prefix hash tree," in *Proceedings of the 23rd annual ACM symposium on Principles of Distributed Computing (PODC '04)*, pp. 368–368, ACM, New York, NY, USA.

[38] A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: supporting scalable multi-attribute range queries," in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '04)*, pp. 353–366, September 2004.

[39] M. Cai, M. Frank, J. Chen, and P. Szekely, "MAAN: a multi-attribute addressable network for grid information services," *Journal of Grid Computing*, vol. 1, pp. 3–14, 2004.

[40] C. Schmidt and M. Parashar, "Squid: enabling search in DHT-based systems," *Journal of Parallel and Distributed Computing*, vol. 68, no. 7, pp. 962–975, 2008.

[41] Y. Chawathe, S. Ramabhadran, S. Ratnasamy, A. LaMarca, S. Shenker, and J. Hellerstein, "A case study in building layered DHT applications," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '05)*, pp. 97–108, ACM, New York, NY, USA.

[42] J. K. Lawder and P. J. H. King, "Using space-filling curves for multi-dimensional Indexing," in *Proceedings of the 17th British National Conferenc on Databases: Advances in Databases (BNCOD '00)*, Springer, London, UK, 2000.

[43] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[44] Bollen et al., Sea and Air Container Track and Trace Technologies: Analysis and Case Studies, Project NO. TPTT01/2002T, APEC, July 2004, http://www.apec-tptwg.org.cn/new/archives/tpt-wg24/safe/its/itf-track-trace.pdf.

[45] IETF Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information, Request for Comments: 3825, http://www.ietf.org/rfc/rfc3825.txt.

[46] Y. Tang, S. Zhou, and J. Xu, "LIGHT: a query-efficient yet low-maintenance indexing scheme over DHTs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 1, pp. 59–75, 2010.

[47] E. Jaffe and J. Albrecht, "PlanetLab—P2P testing in the wild," in *Proceedings of the 9th International Conference on Peer-to-Peer Computing (IEEE P2P '09)*, pp. 83–84, September 2009.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 675605, 12 pages
doi:10.1155/2012/675605

Research Article

Route Anomaly Detection Using a Linear Route Representation

Wen-Chen Hu,[1] Naima Kaabouch,[2] Hung-Jen Yang,[3] and S. Hossein Mousavinezhad[4]

[1] *Department of Computer Science, The University of North Dakota, Grand Forks, ND 58202, USA*
[2] *Department of Electrical Engineering, The University of North Dakota, Grand Forks, ND 58202, USA*
[3] *Industrial Technology Educational Department, National Kaohsiung Normal University, Kaohsiung City 80201, Taiwan*
[4] *Department of Electrical Engineering, Idaho State University, Pocatello, ID 83209, USA*

Correspondence should be addressed to Wen-Chen Hu, wenchen@cs.und.edu

Received 18 March 2012; Accepted 2 June 2012

Academic Editor: MoonBae Song

A location-based service is a service based on the geographical position of a mobile handheld device like a smartphone. This research proposes location-based research, which uses location information to find route anomalies, a common problem of daily life. For example, an alert should be generated when a deliveryman does not follow his regular route to make deliveries. Different kinds of route anomalies are discussed and various methods for detecting the anomalies are proposed in this paper. The proposed method based on a linear route representation finds the matched routes from a set of stored routes as the current route is entered location by location. Route matching is made easy by comparing the current location to linear routes. An alert is generated when no matched routes exist. Preliminary experimental results show the proposed methods are effective and easy to use.

1. Introduction

Table 1 shows the worldwide PC and mobile phone sales according to various market research reports [1]. The number of smartphones shipped worldwide has passed the number of PCs and servers shipped in 2011 and the gap between them is expected to keep bigger. The emerging smartphones have created many kinds of applications that are not possible or inconvenient for PCs and servers, even notebooks. One of the best-seller applications is location-based services (LBSs) according to the following market research.

(i) Hillard [2] reports 80% of smartphone owners have location-based services and half of them use services that provide offers, promotions, and sales based on their current locations.

(ii) The most convenient mobile shopping experience is price comparison and product research according to JiWire [3].

(iii) The number of location-based services users was increased from 12.3 million in 2009 to 33.2 million in 2010 (170% increase) in the US based on SNL Kagan [4].

This paper proposes location-based research, which uses location information to find route anomalies. Different kinds of route anomalies are discussed and various methods for detecting the anomalies are proposed in this research. It is divided into five steps: (i) route data collection, (ii) route data preparation, (iii) route pattern discovery, (iv) route pattern analysis and visualization, and (v) route anomaly detection. The major methods use a technique of incremental location search based on a linear route representation, which facilitates the route storage and matching. It begins the searching as soon as the first location of the search route is entered. Location-by-location, one or more possible matches for the route are found and immediately presented. Route matching is made easy by comparing the current location to linear routes. An alert is generated when no matched routes exist. Preliminary experiment results show the proposed methods are effective and easy to use.

The rest of this paper is organized as follows. Section 2 gives the background information of this research, which includes three themes (i) location-based services, (ii) related location-based research, and (iii) route representations and matching. The proposed system is introduced in Section 3, and several simple methods of route anomaly detection are explained too. Section 4 details the two major methods

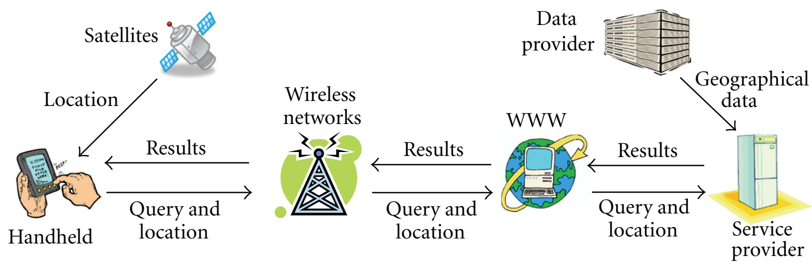


FIGURE 1: A system structure of generic location-based services.

TABLE 1: Worldwide PC, cellphone, and tablet PC sales.

Year	Number of units shipped (million)				
	Mobile phones	PCs and servers	Smart phones	PDAs (without phone capability)	Tablet PCs
2002	432	148	—	12.1	—
2003	520	169	—	11.5	—
2004	713	189	—	12.5	—
2005	813	209	—	14.9	—
2006	991	239	64	17.7	—
2007	1153	271	122	—	—
2008	1220	302	139	—	—
2009	1221	306	166	—	1
2010	1609	346	286	—	17
2011	1775	353	486	—	73

using a linear route representation and incremental location searching. Section 5 gives experimental results and evaluations. The last section gives a summary of this research.

2. Background

Three themes are related to this research: (i) location-based services, (ii) related location-based research, and (iii) route representation and matching, which are discussed in this section.

2.1. Location-Based Services (LBSs). A location-based service is a service based on the geographical position of a mobile handheld device [5, 6]. Two of the LBS examples are (i) finding a nearby ethnic restaurants and (ii) locating a nearby store with the best price of a product. A system structure of location-based services, shown in Figure 1, includes the following five major components [7].

(a) Mobile handheld devices, which are small computers that can be held in one hand. For most cases, they are smartphones.

(b) Positioning system, which is a navigation satellite system that provides location and time information to anyone with a receiver.

(c) Mobile and wireless networks, which relay the query and location information from devices to service providers and send the results from the providers to devices.

(d) Service providers, which provide the location-based services.

(e) Geographical data providers, which are databases storing a huge amount of geographical data such as information about restaurants and gas stations.

An example of location-based services using our research is given step by step as follows.

(1) The smartphone (a) includes an application of finding route anomalies.

(2) A mobile user submits a query of finding route anomalies along with the location information from a positioning system (b) to the application program, which runs in background.

(3) The application program calls the server-side programs (d) located at the Aerospace School of the University of North Dakota along with the location information via mobile or wireless networks (c).

(4) The programs at the servers perform the route anomaly detection using an Oracle database (e) which stores route data. Appropriate actions such as sending an alert are taken when an anomaly occurs.

(5) The results such as an acknowledgement of sending an alert are sent back to the smartphone.

A nice introduction of LBS technologies and standards is given by Wang et al. [8].

2.2. Related Location-Based Research. Various kinds of location-based research can be found in journals and conference proceedings. This subsection discusses several important location-based articles involving travel sequences.

(i) Zheng et al. [9] propose a method to mine interesting locations and travel sequences. Three steps are used in this research.

Step 1. Model multiple individuals' location histories with a tree-based hierarchical graph (TBHG).

Step 2. Based on the TBHG, a HITS- (Hypertext Induced Topic Search) based inference model is constructed.

Step 3. Mine the travel sequences among locations considering the interests of these locations and users' travel experiences.

Finally, a large GPS dataset, collected by 107 users over a period of one year in the real world, is used to evaluate their system. The result shows their HITS-based inference model outperformed baseline approaches like rank-by-count and rank-by-frequency.

(ii) In a paper from Zheng et al. [10], an approach based on supervised learning is proposed to automatically infer transportation mode from raw GPS data. The transportation mode, such as walking and driving, implied in a user's GPS data can provide them valuable knowledge to understand the user. Their approach consists of three parts: (i) a change point-based segmentation method, (ii) an inference model, and (iii) a postprocessing algorithm based on conditional probability. They evaluated the approach using the GPS data collected by 45 users over six months period. The result shows the change point-based method achieved a higher degree of accuracy in predicting transportation modes and detecting transitions between them.

(iii) Previous routes can be used to recommend future travel patterns. Yoon et al. [11] propose itinerary recommendation based on multiple user-generated GPS trajectories. Users only need to provide a start point, an end point, and travel duration to receive an itinerary recommendation. Liu and Chang [12] presents a route recommendation system which guides the user through a series of locations. Their system uses the methods of sequential pattern mining to extract popular routes from a set of stored routes from previous users. It then recommends routes by matching the user's current route with the extracted routes.

Some related location-based research can be found from the articles [13–18].

2.3. Route Representations and Matching. Traditionally, a travel route is stored as a series of locations (latitude and longitude) and route matching uses simple comparison. This research saves the routes as sequences of line segments and the route matching becomes finding the distance between the current location and line segments.

2.3.1. Route Representations. Route representations in computer are similar to image representations because each consists of a set of locations/pixels on a two-dimensional plane. Therefore, the representations of images can be applied to route representations and matching.

(i) Chang et al. [4] proposed a 2D string representation. A matching query may specify a 2D string, transforming retrieval into a 2D subsequence matching.

(ii) A 2D C-string for spatial knowledge representation, which employs a cutting mechanism and a set of spatial operators, was proposed by Lee and Hsu in 1990 [19].

(iii) Huang and Jean [20] proposed a 2D C^{+}-string representation scheme which extended the 2D C-string by including relative metric information about the picture into the strings.

Other representations can be found from the articles by Lee et al. [21] and Wu and Chang [22].

2.3.2. Route Matching. Incremental search is a progressive search, which finds matched text as the search string is entered character by character. Most incremental searches are based on the research of Aho and Corasick [23], who develop an algorithm to locate all occurrences of any of a finite number of keywords in a string of text. The algorithm consists of constructing a finite state pattern matching machine from the keywords and then using the pattern matching machine to process the text string in a single pass. Construction of the pattern matching machine takes time proportional to the sum of the lengths of the keywords. The number of state transitions made by the pattern matching machine in processing the text string is independent of the number of keywords. Several incremental search methods have been proposed as follows.

(i) Meyer [24] gives a problem of searching a given text for occurrences of certain strings, in the particular case where the set of strings may change as the search proceeds. He modified the algorithm of Aho and Corasick to allow incremental diagram construction, so that new keywords may be entered at any time during the search. The incremental algorithm presented essentially retains the time and space complexities of the nonincremental one.

(ii) The machine of Aho and Corasick must be reconstructed all over again when a keyword is appended. Tsuda et al. [25] propose an efficient algorithm to append a keyword for the machine. They present the time efficiency comparison with the original algorithm using the actual simulation results. The simulation results show the speed up factor, by the algorithm proposed, to be between 25- and 270-fold when compared with the original algorithm by Aho and Corasick which requires the reconstruction of the entire machine AC.

(iii) Koenig et al. [26] develop a search algorithm that combines incremental and heuristic search, namely, Lifelong Planning A* (LPA*). It is named "Lifelong Planning" because it reuses information from previous searches. Their method repeatedly finds shortest paths from a given start vertex to a given goal vertex while the edge costs of a graph change.

A survey of incremental heuristic searches can be found from the article by Koenig et al. [27].

3. The Proposed System

The GPS (global positioning system) function of smartphones provides location information of mobile users. Collections of location information are able to depict the mobile users' travel routes such as walking routes between homes and schools or salesman's delivery routes. This research uses the location information to find any route anomalies, for example, a pupil does not take the daily route to school. The proposed system is introduced in this section.

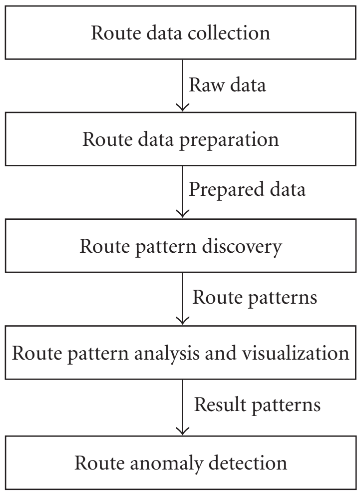


Figure 2: The five steps used by the proposed system.

3.1. The Proposed Steps. This research is to find route anomalies. It is divided into five steps as shown in Figure 2.

(a) *Route data collection*: this step collects route data before the application is used.

(b) *Route data preparation*: raw GPS data is usually not reliable and consistent and includes many noises. It has to be prepared before used.

(c) *Route pattern discovery*: not all routes are valid, for example, a very short route is usually not useful. This step puts valid routes into a database and removes invalid routes.

(d) *Route pattern analysis and visualization*: it analyzes the routes and allows users to view the routes on maps.

(e) *Route anomaly discovery*: this step is used to find any route anomalies, the theme of this research. The method of incremental search is used in this step.

Figure 3(a) shows the application icon, Route Checking, on an Android device. After clicking the icon, it displays the entry interface of this system as in Figure 3(b), which includes three radio buttons.

(i) *Collect route data*, which is for route data collection after this button is submitted. This function can be activated anytime, but most likely it is activated at the beginning of using this application.

(ii) *Check routes*, which redirects to the interface in Figure 9(b) for route checking after this button is submitted.

(iii) *Show stored routes*, which is used to display the information of the stored routes. An example of the basic route information is shown in Table 2, which includes the number and start and end times and locations of each route. The latitude and longitude of a location are represented by r and θ of the polar coordinate system. Routes can be added to the system from time to time and users are able to delete undesirable or never-used routes. Table 2 shows the basic information about routes. Details of each route are stored elsewhere.

Table 2: Basic route information including route numbers, and start and end times and locations.

Route number	Location	Time	Latitude	Longitude
1	Start	05/26/2012, 18:23:42	47.926823°	−97.080351°
	End	05/26/2012, 18:45:23	47.926825°	−97.080351°
2	Start	05/27/2012, 09:31:58	47.926810°	−97.080325°
	End	05/27/2012, 09:50:41	47.926812°	−97.080324°
⋮	⋮	⋮	⋮	⋮
n	Start	05/31/2012, 13:13:37	47.926857°	−97.080357°
	End	05/31/2012, 14:20:37	47.926859°	−97.080358°

3.2. System Implementation. The collected route data is usually raw because GPS data is usually not reliable and consistent and contains many noises [16]. The data needs to be processed before being used effectively. The methods of data preparation include filtering, recovery, restoration, and trajectory. Route data preparation used in this research includes the following.

(i) Two locations with slightly differences are treated the same, for example, the location $(r_1, \theta_1) = (5603, 243.10^\circ)$ is the same as the location $(r_2, \theta_2) = (5609, 243.10^\circ)$, because of low GPS accuracy and the traveler may take the same route but with different locations such as walking on the other side of a street.

(ii) Similar locations in a row (e.g., the traveler is idle) are reduced to one.

(iii) If the distance between two consecutive locations is greater than a threshold value, it may imply a GPS disconnection. Location trajectory may need to be used to insert locations between the two locations [28].

(iv) If a location is substantially different from the surrounding locations, it may imply a GPS noise. The location may be removed and location trajectory is used.

The system then checks the prepared route data and removes invalid routes, which may include the following.

(i) A route is with a very short distance such as few yards long.

(ii) A route is with many broken locations or noises and it cannot be fixed by the method of trajectory.

(iii) A route is with a very long or short duration, for example, less than one minute or longer than two hours.

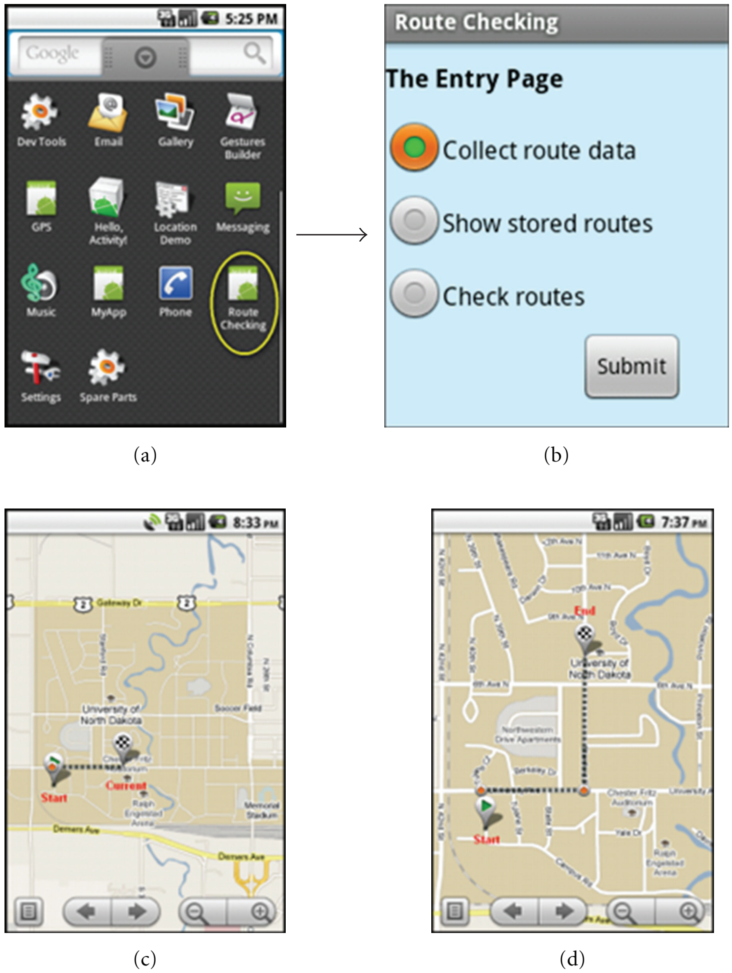


Figure 3: (a) The application icon "Route Checking" on a device, (b) the entry page of the application, (c) the current route map, and (d) an example of a stored route.

This research uses location information to detect route anomalies. Many methods can be used to find route anomalies. Four kinds of detection are introduced in this subsection: (i) time check, (ii) border check, (iii) start and destination check, and (iv) route check. The first three methods are simple and are explained in this subsection. The last method, route check, is the focus of this research and is more complicated. It uses the method of pattern matching to find any route anomalies. The next subsection will be dedicated to it.

(i) *Time check*: for example, the Route number 1 takes about 23 minutes. If a trip follows the route and takes far more than 23 minutes, then an alert may be generated. Also, the start and end times can be used in this check. For example, the schools start at 8 : 30 AM. If the student has not arrived at school by 8 : 30 AM, then an alert may be generated.

(ii) *Border check*: for example, the delivery routes are within a community. If a route reaches out of the community, then an alert may be generated. An easy way to find the border is to box the routes such as the one shown in Figure 4.

(iii) *Start and destination check*: if the traveler does not start from any beginning locations of routes or does not reach any destination by a specific time, it may deserve an alert.

(iv) *Route check*: the above methods are simple, but lack accuracy. This method checks the routes with higher accuracy and uses more complicated algorithms, which will be detailed in the next section.

4. Route Checking Using a Linear Route Representation

Route checking is more complicated. This section discusses the methods used to find route anomalies. Traditional routes are represented by series of locations, which are complex and difficult to use. An algorithm is developed to straighten the routes so the routes can be stored as a set of line segments and route matching becomes a simple task of checking the distance between the current location and line segments. The proposed route checking can be divided into two cases, ordered and unordered routes. There is no alert generated if the traveler does not start at the beginning location or stop

Figure 4: An example of boxing a route.

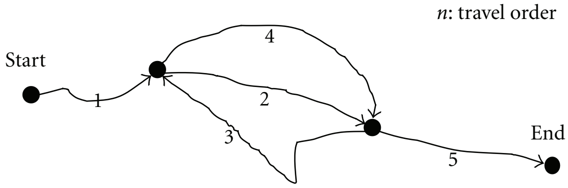


Figure 5: An example of order routes.

at the end of a route since the start and destination check can be easily used to check this condition.

4.1. Linear Approximating a Human Travel Route. The proposed linear approximation algorithm converts a human travel route into line segments. The approximation captures the essence of a route in the fewest possible line segments. A polygonal approximation, based on an error function, is applied to this method [29]. Before applying this algorithm, however, the route has to be smoothed to unit thickness so that branch routes may be located. Let e be the maximum allowable error. For a given location A, through which an approximation line must pass, one can define two points B and C at a distance e from A. The algorithm searches for the longest segment where the curve is contained between two parallel tangents starting B and C (see Algorithm 1).

The function *BEST_PATH* finds the route of the longest line segment when the location p_i has more than one 8-neighbor. This is why the route needs smoothing before applying the algorithm. A nonunit thickness route may mislead the algorithm into calling the *BEST_PATH* function. This algorithm requires quadratic time because the *BEST_PATH* function needs to check as many paths as possible and any location on the route, may invoke it. Figure 5 shows an example of order routes and Figure 6 shows the corresponding linear route after applying the algorithm *LINEAR_APPROX* to the route in Figure 5.

4.2. Ordered Routes. For this kind of routes, the order of the locations is relevant, for example, bus routes. An example of ordered routes is given in Figure 5, where the directional subroutes are the numbers 1, 2, 3, 4, and 5. The numbers

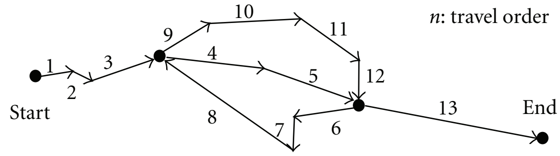


Figure 6: The corresponding linear route after applying the algorithm LINEAR_APPROX to the route in Figure 5.

also imply the travel order, for example, the traveler takes the route: start → 1 → 2 → 3 → 4 → 5 → end. Assume the traveler must travel along the stored routes. A simple algorithm using incremental location searching for finding route anomalies in ordered routes is given below. It checks the traveler's locations one by one. If the distance between the current location and the current line segments is greater than a threshold value, then it reports an anomaly. The traveler's beginning location can start anywhere in a route and the traveler can end the checking anywhere and anytime. (see Algorithm 2).

Assume there are five available routes where the letters represent line segments:

(1) `c → a → e → f → n → l → s → p → g → h`

(2) `c → a → d → e → f → m → n → l → s → t → p → g → h → j`

(3) `s → t → o → g → h → c → a → d → f → m → n`

(4) `n → l → s → t → p → g → h → l → s → t → o → p`

(5) `c → a → d → b → e → m → r → n → s → t → g → h`

Two examples are given next to show how this algorithm works.

(a) This example shows a route does not generate alerts by using this algorithm. Assume a traveler takes the route `c → c → a → d → d → d → f → l → l`. The five stored routes after applying the algorithm are as follows:

(1) `c → a → e → f → n → l → s → p → g → h` (2 matches)

(2) `c → a → d → e → f → m → n → l → s → t → p → g → h → j` (perfect matches)

(3) `s → t → o → g → h → c → a → d → f → m → n` (4 matches)

(4) `n → l → s → t → p → g → h → l → s → t → o → p` (no match)

(5) `c → a → d → b → e → m → r → n → s → t → g → h` (3 matches)

where an underlined location means a match. The Route number 2 has a perfect match, so no alert is generated by this algorithm. However, an alert may be generated because no end location of any route is reached. This anomaly can be found by the start and destination check as mentioned in Section 4.1.

// Linear Approximating a Human Travel Route

LINEAR_APPROX (ROUTE, p_i, p_j, e)
  // ROUTE: a human travel route in a series of locations (latitude, longitude)
  // p_i: the initial location of the route
  // p_j: the neighbor of p_i on the route
  // e: the maximum allowable error

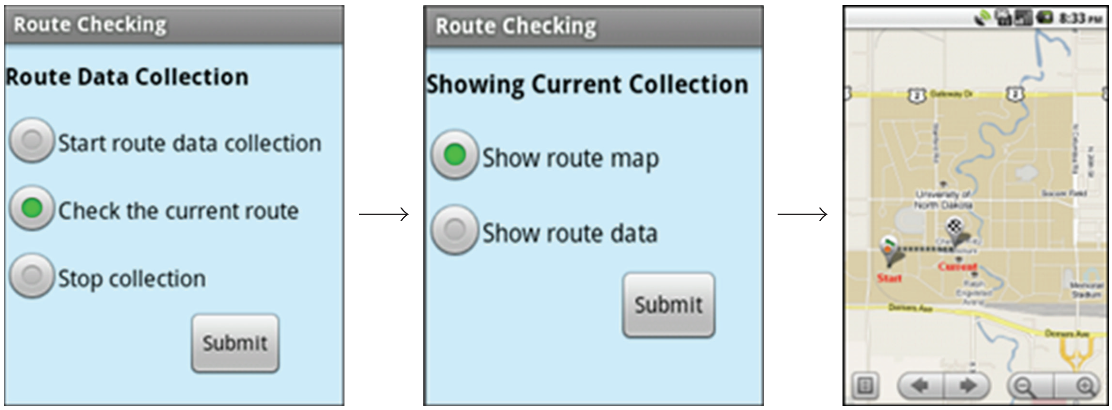
$b \leftarrow p_i + e$
$c \leftarrow p_i - e$
IS_FIRST ← TRUE
while NUMBER_8_NEIGHBOR (p_i) $\neq 0$
  while NUMBER_8_NEIGHBOR (p_i) > 1
    $p_j \leftarrow$ BEST_PATH (ROUTE, p_i, e)
    LINEAR_APPROX (ROUTE, p_i, p_j, e)
    print p_i
  ROUTE [p_i] ← CLEAR
   $\overline{L_b} \leftarrow \overline{bp_j}$
   $\overline{L_c} \leftarrow \overline{cp_j}$
   if IS_FIRST
     $\overline{L_{\text{above}}} \leftarrow \overline{L_b}$
     $\overline{L_{\text{below}}} \leftarrow \overline{L_c}$
     IS_FIRST ← FALSE
 else
   if $\overline{L_b}$ **is above** $\overline{L_{\text{above}}}$
     $\overline{L_{\text{above}}} \leftarrow \overline{L_b}$
   if $\overline{L_c}$ **is below** $\overline{L_{\text{below}}}$
     $\overline{L_{\text{below}}} \leftarrow \overline{L_c}$
   if $\angle(\overline{L_{\text{above}}}, \overline{L_{\text{below}}}) > 0$
     print p_i
     $b \leftarrow p_i + e$
     $c \leftarrow p_i - e$
     IS_FIRST ← TRUE
 $p_i \leftarrow p_j$
 $p_j \leftarrow$ 8_NEIGHBOR(p_i)
ROUTE[p_i] ← CLEAR
print p_i

ALGORITHM 1

Route Checking for Ordered Routes

(1) Every stored route is an available route.
(2) Find the start location of the traveler in an available route i and set the current line segment i_c to the first matched segment.
(3) d_0 = the distance between the current location and the current line segment i_c.
d_1 = the distance between the current location and the next line segment i_{c+1}.
if [$d_0 \leq e$ (the allowable error)], then do nothing,
else if [($d_0 > e$) and ($d_1 \leq e$)], then $i_c = i_{c+1}$ and $i_{c+1} = i_{c+2}$ (the next segment after i_{c+1}),
else remove the route from the available routes. If the available routes are empty, report an anomaly and stop.
(4) Repeat the above step for the next location of the traveler until the traveler ends the route checking.

ALGORITHM 2



(a) (b) (c)

Figure 7: (a) The page of route data collection, (b) the page of checking the current route, and (c) the current route map.

(b) This example shows a route generates an alert by using this algorithm. Assume a traveler takes the route c → a → d → d → f → p → p → p. The five stored routes after applying the algorithm are as follows:

(1) c → a → e → f → n → l → s → p → g → h (2 matches)

(2) c → a → d → e → f → m → n → l → s → t → p → g → h → j (4 matches)

(3) s → t → o → g → h → c → a → d → f → m → n (4 matches)

(4) n → l → s → t → p → g → h → l → s → t → o → p (no match)

(5) c → a → d → b → e → m → r → n → s → t → g → h (3 matches)

The last location p does not have a match in any route so an alert is generated by this algorithm.

4.3. Unordered Routes. For unordered routes, the order of the locations, maybe except the start and end locations, is irrelevant, for example, newspaper delivery. Figure 7 shows an example of unordered routes if the numbers are ignored. In order to find anomalies in unordered routes, the route data collection needs to find and save all line segments connected to intersections, such as the {3, 4, 8, 9} and {5, 6, 12, 13} in Figure 10, and bidirectional subroutes among the start and end segments of each route. An intersection is a road junction where two or more roads either meet or cross at grade. For example, the route in Figure 10 includes the following items:

(i) Start segment: Segment 1,

(ii) End segment: Segment 13,

(iii) Line groups: {3, 4, 8, 9} and {5, 6, 12, 13},

(iv) Bidirectional subroutes: 1↔2↔3, 4↔5, 6↔7↔8, 9↔10↔11↔12, and 13.

The algorithm of finding route anomalies in unordered routes is given below. It is more complicated compared to the one for ordered routes because the traveler's next location could be in several possible subroutes. When an intersection is reached, the algorithm checks all available subroutes from the intersection.

Algorithm 3 does not check whether all subroutes are visited. It could be easily remedied by adding a checker on each subroute. Also, the algorithm generates an alert if a subroute is visited twice because it is unusual to travel a subroute twice in most cases. It could be easily modified if users do not want it to generate an alert for this condition. Again, one example is given next to show how this algorithm works. Assume there are three available routes where the letters represent line segments. The three routes and the items saved for each route are listed in the following:

Route 1. **c → b → d → a → e → f → n → l → t → p → g → i → h**

(I) Start segment: Segment c,

(II) End segment: Segment h,

(III) Line groups: {a, e, p, t} and {g, i, l, n},

(IV) Subroutes: (i) c↔b↔d↔a, (ii) e↔f↔n, (iii) l↔t, (iv) p↔g, and (v) i↔h.

Route 2. **c → b → d → a → p → g → o → r → n → j → l → t → s → u**

(I) Start segment: Segment c,

(II) End segment: Segment u,

(III) Line groups: {a, p, s},

(IV) Subroutes: (i) c↔b↔d↔a, (ii) p↔g↔o↔r↔n↔j↔l↔t, and (iii) s↔u.

Route 3. **c → d → a → e → y → f → z → n → k → i → h**

(i) Start segment: Segment c,

(ii) End segment: Segment h,

(iii) Line groups: None,

(iv) Subroutes: c↔d↔a↔e↔y↔f↔z↔n↔k↔i↔h.

Route Checking for Unordered Routes

(1) Every stored route is an available route and every sub-route of an available route is an available sub-route.
(2) Find the start location of the traveler in an available sub-routes of an available route i and set the current line segment i_c to the first matched segment.
(3) d_0 = the distance between the current location and the current line segment i_c.
d_1 = the shortest distance between the current location and the line segment i_{c+1} among the ones connected to i_c.
if $[d_0 \le e$ (the allowable error)], then do nothing,
else if $[(d_0 > e)$ and $(d_1 \le e)]$, then mark the i_c unavailable and $i_c = i_{c+1}$,
else remove the route from the available routes. If the available routes are empty, report an anomaly and stop.
(4) Repeat the above step for the next location of the traveler until the traveler ends the route checking.

Algorithm 3

The following example shows a route generates an alert by using this algorithm. Assume a traveler takes the route `a` → `e` → `f` → `f` → `m` → `n` → `n` → `i`. The following list shows how the algorithm works location by location:

(1) **a** → `e` → `f` → `f` → `m` → `n` → `n` → `i`, where the current location is `a`:

Route 1: `c`↔`b`↔`d`↔<u>`a`</u>
Route 2: `c`↔`b`↔`d`↔<u>`a`</u>
Route 3: `c`↔`d`↔<u>`a`</u>↔`e`↔`y`↔`f`↔`z`↔`n`↔`k`↔`i`↔`h`

(2) `a` → **e** → `f` → `f` → `m` → `n` → `n` → `i`, where the current location is `e`:

Route 1: `e`↔`f`↔`n`
Route 2: none
Route 3: `c`↔`d`↔`a`↔<u>`e`</u>↔`y`↔`f`↔`z`↔`n`↔`k`↔`i`↔`h`

(3) `a` → `e` → **f** → `f` → `m` → `n` → `n` → `i`, where the current location is `f`:

Route 1: `e`↔<u>`f`</u>↔`n`
Route 2: none
Route 3: `c`↔`d`↔`a`↔`e`↔`y`↔<u>`f`</u>↔`z`↔`n`↔`k`↔`i`↔`h`

(4) `a` → `e` → `f` → **f** → `m` → `n` → `n` → `i`, where the current location is `m`:

Route 1: `e`↔<u>`f`</u>↔`n`
Route 2: none
Route 3: `c`↔`d`↔`a`↔`e`↔`y`↔<u>`f`</u>↔`z`↔`n`↔`k`↔`i`↔`h`

(5) `a` → `e` → `f` → `f` → **m** → `n` → `n` → `i`, where the current location is `m`:

Route 1: none
Route 2: none
Route 3: none

where an underlined segment means a match. An alert is generated because there are no available subroutes when the current location is `m`.

Table 3: An example of detail information of a route.

number (minute)	Time	Latitude	Longitude
0	05/26/2010, 18:23:42	47.926823°	−97.080357°
1	05/26/2010, 18:24:42	47.926823°	−97.080357°
⋮	⋮	⋮	⋮
22	05/26/2010, 18:45:23	47.926825°	−97.080357°

5. Experimental Results

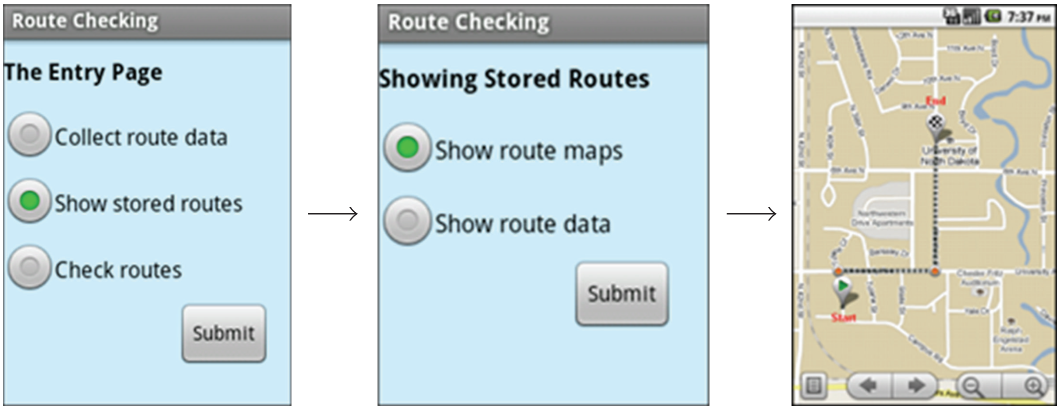
This section gives the experimental results.

5.1. Route Data Collection. The first step of this research is to collect route information. The collection could be activated anytime and anywhere. Figure 3(b) shows the entry page of the application. After a user picks the button "collect route data," the system displays the interface in Figure 7(a) including the following three radio buttons.

(i) *Start collecting route data*, which starts collecting route data. Typical location information provided by a smartphone includes times, the latitude, and longitude of a location. The location information is collected as frequently as possible and the collection frequencies depend on the travelling methods. For example, the frequencies for walking, biking, and driving are different. This process runs in background by using multithreading, so the smartphone can still function as usual.

(ii) *End collection*, which ends the current route data collection.

(iii) *Check the current route*, which is used to check the status of the current data collection including the map as in Figure 7(c) and data as in Table 3.



(a) (b) (c)

Figure 8: (a) The system entry page, (b) the page of showing stored routes, and (c) an example of a stored route.

Table 2 shows the basic information about routes. Details of each route are stored elsewhere. An example of the Route number 1 details is given in Table 3, where the locations and times are collected periodically, for example, every minute for walking and every 10 seconds for driving.

5.2. Route Data Preparation and Route Pattern Discovery, Analysis, and Visualization. The collected route data is usually raw because GPS data is usually not reliable and consistent and contains many noises [13]. The system also allows users to check the stored routes as in Figure 8(b). Other than showing them the route information as in Tables 2 and 3, users will also like to view the stored routes. One of the routes is shown in Figure 8(c).

5.3. Route Anomaly Detection. This research uses location information to detect route anomalies. An alert as in Figure 9(c) is generated when an anomaly is found. Otherwise, the smartphone just functions as usual. The alert can be sent via an email or a phone call. The interface in Figure 9(b) is for checking routes including the following three radio buttons.

(i) *Start checking the route*, which runs in background by using multithreaded programming, so the smartphone can still function as usual.

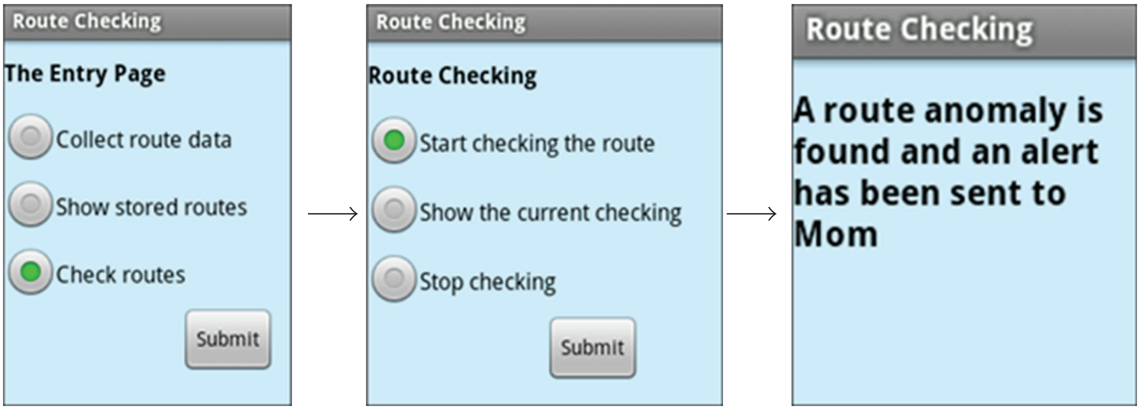
(ii) *End checking*, which stops the current route checking. Also, the check ends when an anomaly is found.

(iii) *Show the current checking*, which is used to check the status of the current route checking. For example, how closely will the current route trigger an alert or how many routes are matched so far? Other than showing the data of the current route as in Table 3 by using the bottom button in Figure 10(b), the system also shows the current position in a possible route as in Figure 10(c) by using the top button in Figure 10(b).

The proposed methods are convenient and effective. The execution is also efficient. The times used for the time check and start and destination check are constant. The times for the border check and route checks are $O(n)$, where n is the number of user locations, because the checks are performed location by location and each location requires a constant time to do the matching. The time for the procedure *Linear_Approx* is also $O(n)$, where n is the number of locations in a route, because the algorithm straightens the route location by location. Additionally, the procedure is only used during route collections, but not route checking, which happens more often.

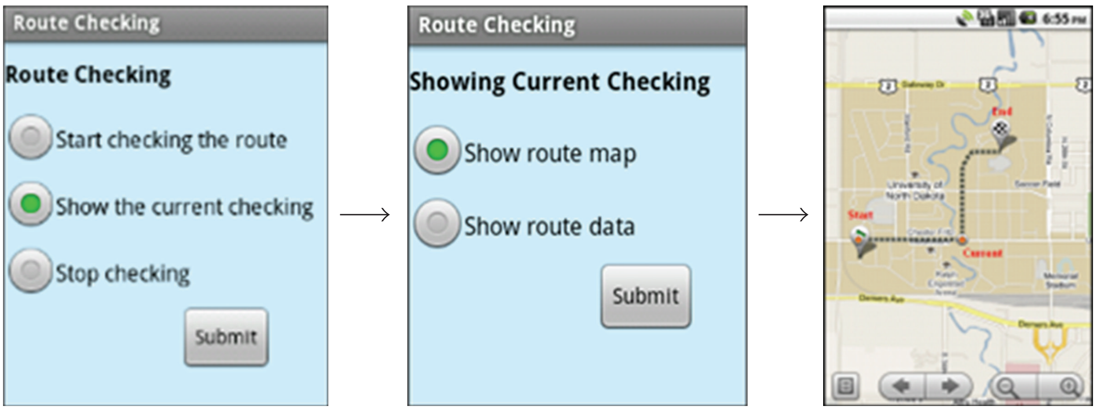
6. Conclusion

Mobile application stores (or app stores) sell or provide mobile applications/services for handheld devices such as smartphones. The applications/services are not necessarily from the storeowners. Many of them are from the third parties such as independent developers. A wide variety of mobile applications is available on the stores. One of the best-selling apps is related to location-based services, which cover a wide range of usages such as finding the lowest-price product in a nearby store and locating the nearest gas stations. This research proposes location-based research, which uses location information to find route anomalies, a common problem of daily life. For example, an alert should be generated when a school bus misses part of a route. Different kinds of route anomalies are discussed and various methods for detecting the anomalies are proposed in this paper. It is divided into five steps: (i) route data collection, (ii) route data preparation, (iii) route pattern discovery, (iv) route pattern analysis and visualization, and (v) route anomaly detection. The major methods use a linear route representation and incrementally search locations, which finds matched routes as the search route is entered location by location. It begins the searching as soon as the first location of the search route is entered. Location by location, one or more possible matches for the route are found and immediately presented. An alert is generated when no matched routes exist. Experimental results show the proposed methods are effective and easy to use.



(a) (b) (c)

Figure 9: (a) The entry page of the application, (b) the page of route checking, and (c) an acknowledgment message after detecting a route anomaly.



(a) (b) (c)

Figure 10: (a) The page of route checking, (b) the page of showing cu rrent checking, and (c) the current position in a stored route.

Other than the linear route representation, the proposed incremental location search is based on string matching, which is simple but effective. A search based on the following methods is worth consideration.

(i) *Finite automata*: the collected routes are used to build a finite automaton, which is then used to check any route anomalies.

(ii) *Matrix multiplication*: similar routes are found by matrix multiplications between the current route and the stored routes. If the product is greater than a threshold value, the stored route is said to be *similar* to the current route. Since the matrices are stored as strings, the multiplications can be done efficiently.

(iii) *Neural networks*: a route is a sequence of locations. Route matching is used to find any route anomalies and a modified Hopfield neural network can be designed to solve this problem.

(iv) *Approximate string matching*: routes are stored as strings or sequences of locations. Approximate string matching is then used to find any route anomalies.

References

[1] W.-C. Hu, "Worldwide: Computer and Device Sales," 2012, http://www.handheldresearch.org/.

[2] F. Hillard, "Digital Influence Study," 2012, http://www.factbrowser.com/facts/4671/.

[3] JiWire, "JiWire Mobile Audience Insights Report Q2 2011," 2011, http://www.factbrowser.com/facts/3196/.

[4] C. C. Chang, E. Jungert, and G. Tortora, *Intelligent Image Database System*, World Scientific, Singapore, 1996.

[5] K. Kolodziej and J. Hjelm, *Local Positioning Systems: LBS Applications and Services*, CRC Taylor & Francis, 2006.

[6] A. Kupper, *Location-Based Services: Fundamentals and Operation 2005*, John Wiley & Sons.

[7] S. Steiniger, M. Neun, and A. Edwardes, "Foundations of Location-Based Services," 2006, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.1844.

[8] S. Wang, J. Min, and B. K. Yi, "Location based services for mobiles: technologies and standards," in *Proceedings of the IEEE International Conference on Communication (ICC '08)*, Beijing, China, May 2008.

[9] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in

Proceedings of the 18th International World Wide Web Conference (WWW '09), Madrid, Spain, April 2009.

[10] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 247–256, April 2008.

[11] H. Yoon, Y. Zheng, X. Xie, and W. Woo, "Smart itinerary recommendation based on user-generated GPS trajectories," *Lecture Notes in Computer Science*, vol. 6406, pp. 19–34, 2010.

[12] D. Liu and M. Chang, "Recommend touring routes to travelers according to their sequential wandering behaviours," in *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN '09)*, pp. 350–355, Kaohsiung, Taiwan, December 2009.

[13] M. Duckham, S. Winter, and M. Robinson, "Including landmarks in routing instructions," *Journal of Location Based Services*, vol. 4, no. 1, pp. 28–52, 2010.

[14] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pp. 330–339, San Jose, Calif, USA, August 2007.

[15] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xie, "Discovering spatio-temporal cal interactions in traffic data streams," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, San Diego, Calif, USA, August 2011.

[16] G. Myles, A. Friday, and N. Davies, "Preserving privacy in environments with location-based applications," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 56–64, 2003.

[17] H. Tootell, "Location-based services and the price of security," in *Proceedings of the IEEE Internaional Symposium on Technology and Society (ISTAS '06)*, Queens, NY, USA, June 2006.

[18] L. X. Pang, S. Chawla, W. Liu, and Y. Zheng, "On mining anomalous patterns in road traffic streams," in *Proceedings of the 7th International Conference on Advanced Data Mining and Applications (ADMA '11)*, pp. 237–251, Beijing, China, December 2011.

[19] S. Y. Lee and F. J. Hsu, "2D C-string: a new spatial knowledge representation for image database systems," *Pattern Recognition*, vol. 23, no. 10, pp. 1077–1087, 1990.

[20] P. W. Huang and Y. R. Jean, "Using 2D C^+-strings as spatial knowledge representation for image database systems," *Pattern Recognition*, vol. 27, no. 9, pp. 1249–1257, 1994.

[21] S. Y. Lee, M. K. Shan, and W. P. Yang, "Similarity retrieval of iconic image database," *Pattern Recognition*, vol. 22, no. 6, pp. 675–682, 1989.

[22] T. C. Wu and C. C. Chang, "Application of geometric hashing to iconic database retrieval," *Pattern Recognition Letters*, vol. 15, no. 9, pp. 871–876, 1994.

[23] A. V. Aho and M. J. Corasick, "Efficient string matching: an aid to bibliographic search," *Communications of the ACM*, vol. 18, no. 6, pp. 333–340, 1975.

[24] B. Meyer, "Incremental string matching," *Information Processing Letters*, vol. 21, no. 5, pp. 219–227, 1985.

[25] K. Tsuda, M. Fuketa, and J. I. Aoe, "An incremental algorithm for string pattern matching machines," *International Journal of Computer Mathematics*, vol. 58, no. 1-2, pp. 33–42, 1995.

[26] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A∗," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004.

[27] S. Koenig, M. Likhachev, Y. Liu, and D. Furcy, "Incremental heuristic search in AI," *AI Magazine*, vol. 25, no. 2, pp. 99–112, 2004.

[28] Y. Chen, K. Jiang, Y. Zheng, C. Li, and N. Yu, "Trajectory simplification method for location-based social networking services," in *Proceedings of the International Workshop on Location Based Social Networks (LBSN '09)*, pp. 33–40, 2009.

[29] T. Pavlidis, *Structural Pattern Recognition*, Spring, New York, NY, USA, 1977.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 963728, 11 pages
doi:10.1155/2012/963728

Research Article

Context-Aware Adaptation of Component-Based Systems: An Active Repository Approach

Sindolfo Miranda Filho,[1] Julio Melo,[1] Luiz Eduardo Leite,[2] and Guido Lemos[3]

[1] *Centro de Tecnologia, Universidade Federal do Rio Grande do Norte, 59078-900 Natal, RN, Brazil*
[2] *Escola de Ciência e Tecnologia, Universidade Federal do Rio Grande do Norte, 59078-900 Natal, RN, Brazil*
[3] *Departamento de Informática, Universidade Federal da Paraíba, 58051-900 Paraíba, PB, Brazil*

Correspondence should be addressed to Sindolfo Miranda Filho, sindolfo@lavid.ufpb.br

Received 17 March 2012; Revised 17 June 2012; Accepted 18 June 2012

Academic Editor: Goreti Marreiros

Context-aware systems are able to monitor and automatically adapt their operation accordingly to the execution context in which they are introduced. Component-based software engineering (CBSE) focuses on the development and reuse of self-contained software assets in order to achieve better productivity and quality. In order to store and retrieve components, CBSE employs component repository systems to provide components to the system developers. This paper presents an active component repository that is able to receive the current configuration from the context-aware system and compute the components and the new architecture that better fit the given context. Since the repository has a wide knowledge of available components, it can better decide which configuration is more suitable to the running system. The repository applies Fuzzy logic algorithm to evaluate the adequacy level of the components and GRASP algorithm to mount the new system architecture. In order to verify the feasibility of our approach, we use a digital TV middleware case study to achieve experimental results.

1. Introduction

Pervasive computing is becoming increasingly popular, as introduced by Weiser [1], the term pervasive refers to the seamless integration of computer devices and software into the everyday life. Context-awareness adaptation is an important field of the pervasive computing area. Context-aware systems are able to monitor and automatically adapt their operation accordingly to the execution context in which they are introduced.

Component-based software engineering (CBSE) [2, 3] focuses on the development and reuse of self-contained software assets in order to achieve better productivity [4] and quality as software systems are composed by previously developed components used (and tested) in other projects. In this approach, the software system is composed by self-contained components that explicitly declare their provided functionalities (provided interfaces), required functionalities (required interfaces), and also their execution context requirements.

In addition to the advantages described previously, component-based software (CBS) may present other features like component update, functionality enhancement, and adaptability.

If a CBS needs to be updated, only the specific component implementing the updated feature needs to be updated. Functionality enhancement is also facilitated since new components with new functionalities can be added and dynamically loaded into the system. Finally, adaptability can also be accomplished by configuring and/or replacing a component by another that better fit the current execution environment.

One of the challenges faced by CBSE is the task of discovering suitable reusable assets that fulfill the requirements of the particular software system under development. In order to address such problem, several component repository systems were proposed [5–9]. Accordingly to [10], a component repository is a software system that provides functionalities to locate, select, and retrieve software components.

Ye [11] proposed a repository that provides active mechanisms of information delivery. Ye's active repository is capable of providing information to the users by monitoring their development activities without the need to receive explicit queries from them. These systems execute in background inside an integrated development environment (IDE) monitoring the user activity and suggesting possible software components to be used in the current development context.

Both the traditional and the active repositories provide functionalities only during the development phase of the software system. Once the system is deployed, the repositories are of no use. This way, in order to promote the runtime adaptation of context-aware component-based systems, this paper presents an active repository that actively provides new software components that are more adapted to the context of the adaptable system. Since the repository has a wide knowledge of the available components, it can better decide which configurations of components are more suitable to the running system. In the proposed approach, the context aware system informs to the repository its current configuration and its context information and the repository is able to compute the components and new architecture that better fit the given context. In this way, the component repository commonly adopted in component-based software systems is expanded to provide components, not only during the development stage but also context aware components during the operational stage of the system life cycle.

This paper is organized as follows. Section 2 presents the repository architecture and the mechanism used by our approach in order to evaluate the adequacy level of a component to the given context. Section 3 presents the heuristic algorithms used to compute a new architecture to the given system. These algorithms use the component adequacy level described in Section 2. Section 4 presents our digital TV middleware case study. Section 5 presents some experimental results achieved with our digital TV middleware implementation. The related works are discussed in Section 6. Finally, Section 7 presents our concluding remarks and future directions.

2. System Architecture

To allow the construction of adaptive component-based systems, that will be called now and forth client systems, we have created an architecture named REATIVO that will handle the client system context changes and generate new system architectures over the time through component reconfiguration. Figure 1 shows REATIVO's general architecture.

This architecture is not intended to be restricted to a specific component model. However, to use REATIVO's services, the client system needs to be developed using a component model that allows at least the simple reconfiguration commands (add, remove, replace, connect, or disconnect a component in the system).

In addition, the client system must implement some modules that allow REATIVO retrieve the system current configuration state and context representation.

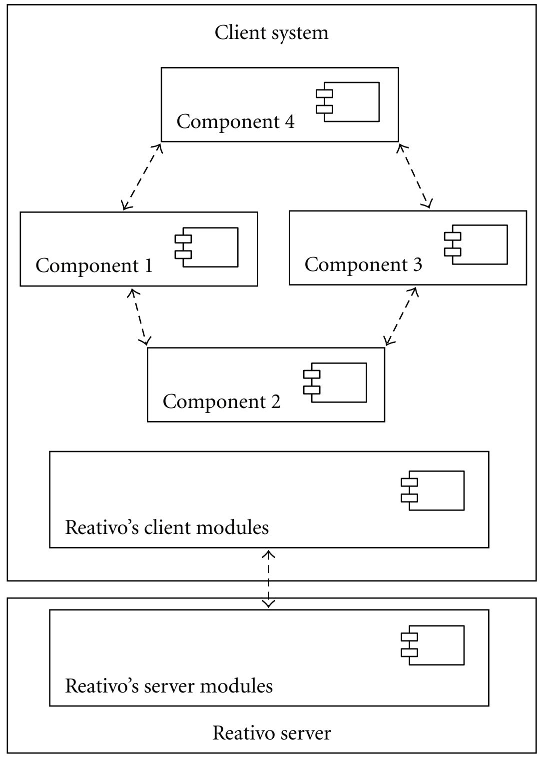


Figure 1: General architecture.

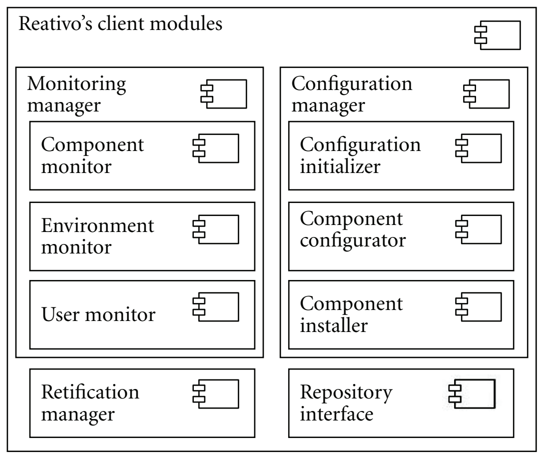


Figure 2: Client components.

REATIVO's modules can be divided into two disjoint groups, the Client components and the Server components. Figures 2 and 3 show these components, respectively. As shown in Figure 2, the Client components are as follows.

(i) Monitoring manager: responsible to monitor the current execution context in terms of the state and component architecture (Component Monitor), the execution environment (Environment Monitor) and the users (User Monitor).

(ii) Reification Manager: responsible to combine information collected by the Monitoring Manager in a

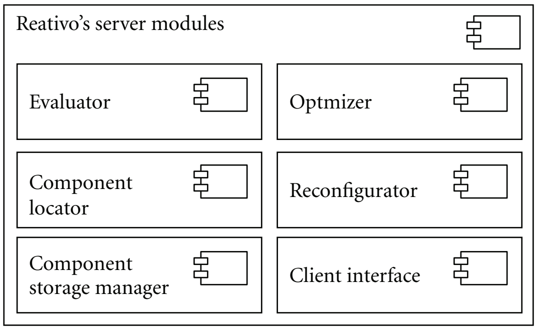


Figure 3: Server components.

meta-representation that will be used to generate a new optimized configuration to the system.

(iii) Configuration Manager: responsible for the instantiation of the initial system architecture (Configuration Initializer) and for instantiate new architectures sent by the REATIVO server, new architectures will be instantiated by adding, removing, or replacing components (Component Installer) or interconnecting existing ones (Component Configurator).

(iv) Repository Interface: provides communication with REATIVO server. This module is able to provide the client system an abstraction of the server location in order to enable the use of the distributed services.

The REATIVO server provides a set of services for receive client system context and provide reconfiguration commands if needed. The server components are listed below.

(i) Evaluator: evaluates quantitatively the adequacy level of a given component accordingly to a given client context.

(ii) Storage Manager: stores components into the repository.

(iii) Component locator: locates components in the distributed repository.

(iv) Optimizer: Implements heuristic algorithms to solve the best configuration problem (See Section 3. Generating optimized component architectures).

(v) Reconfigurator: reconfigures the client system by sending the reconfiguration commands: add, remove, replace, connect, or disconnect components.

(vi) Client Interface: Implements the communication between the clients and the REATIVO server.

2.1. Repository Behavior. When the client system starts up, the Configuration Initializer reads the initial component architecture and initializes it to put the system in full execution.

Once in execution, the system variables will be monitored by REATIVO client components. The variables monitored are chosen in accordance to an ontology developed focusing the specific application domain. When a context modification is detected, the client system generates a meta-representation through the Reification Manager and uses the Repository Interface to pass it to REATIVO's server, which will activate the Optimizer in order to start the process to generate a new architecture.

The Optimizer will execute a heuristic algorithm to compute the best component architecture based on the received context. This heuristic algorithm will be described in Section 3. Generating optimized component architectures and relies on the adequacy level of each component.

When a better architecture is found, REATIVO's server sends the reconfiguration commands necessary to change the client system by using the Client Interface component.

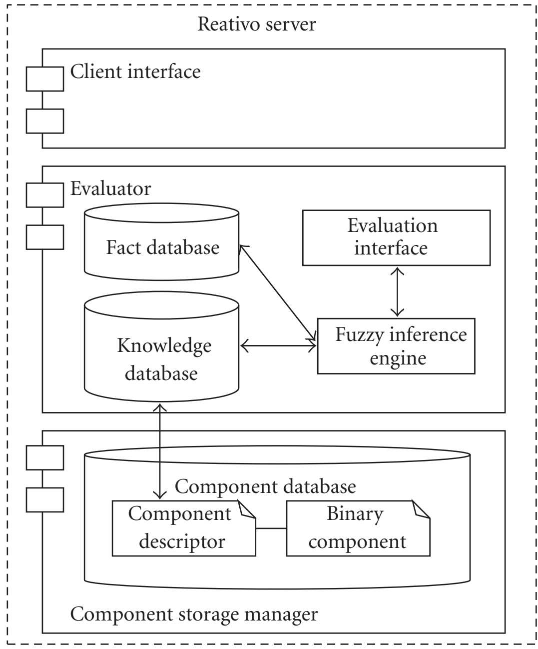


Figure 4: Component evaluation architecture.

2.2. Component Evaluation—Calculating the Component's Adequacy Level. To be able to evaluate a given component for a given execution context, REATIVO uses the Evaluator component (see Figure 4). The implementation of the Evaluator uses Fuzzy Logic [12] to perform a quantitative evaluation of the adequacy level of a component to a given execution context. In this context, the adequacy level consists of values ranging from 0% to 100% of adequacy.

When the Evaluator is queried for a component evaluation it requests to the Storage Manager a component description that contains a set of rules in the format: if "condition" then "component is adequate." By using this modeling one could thought that it is better to use first order logic to decide whenever a component is adequate or not, however some measurements in the execution context may not be like the rule: if "received signal has noise" then "component is adequate." In this case, the signal could have much, moderate or just a little amount of noise, so using first order logic is not a good choice since it could not deal well with inaccurate values.

By using Fuzzy logic in the previous example, we could conclude that the component is "much adequate," "moderate adequate" or "little adequate" by using the generalized modus ponens [12] what explain our choice to the Fuzzy approach.

The Evaluator stores the component description in a Knowledge Database and the execution context sent by the client system in the Fact Base. This way, the Fact Base contains information determined by the Domain Ontology that constrains which are the variables and their value range, in addition to the following parameters to be used by the fuzzy inference engine.

(i) the pertinence function to the fuzzy set which the input variables may belong;

(ii) math operators that implement the fuzzy logic operations;

(iii) defuzzification function to be utilized;

(iv) Fuzzy rules to be inserted in the Knowledge Base;

(v) Fuzzy sets to which the output variables of the Inference Engine may belong.

In this context, the ontology is used to define which concepts are relevant to the domain (these concepts will be monitored by the client system) and the value range of each of these concepts. On the other hand, the component adequacy level is given by the Fuzzy inference engine, which receives as input the values of the domain ontology variables collected by the client system.

Each component in the repository provides its own set of parameters (listed previously) defined in the Component Descriptor and delivered with the component in the Storage Manager. This way the component developer has flexibility to define how the Evaluator module will behave when evaluating the components.

This concept may cause a problem where some components are evaluated differently for the same criteria. For instance a company could specify that its component has an evaluation 50 for determined context and another company could specify that its component, less adequate in fact, has evaluation 60 for the same context. In order to solve such problem, the component description must be specified according to a well defined standard; however, this problem is out of our scope in this work. To simplify the issue, our work assumes that the components are implemented by the same manufacturer or the components are defined by a consortium of manufacturers that uses the same evaluation criteria.

An important aspect that can be noted here is that REATIVO's repository approach is domain independent. If the repository needs to be used with a different client application, within a different application domain, a new domain ontology needs to be developed to the specific domain and the components registered in the repository needs to define their parameters to the Fuzzy inference engine compatible with the new ontology, in other words, needs to take into account the values of the variables defined in the new ontology. No changes need to be done in the repository server logic.

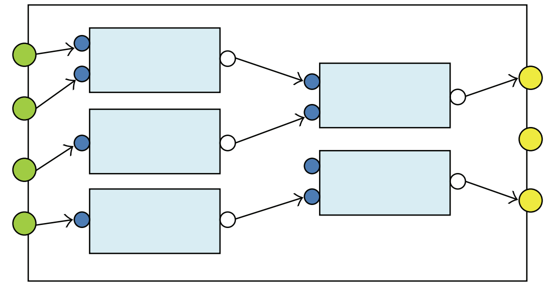


Figure 5: Internal diagram of a general component based system.

3. Generating Optimized Component Architectures

REATIVO stores a set of components that needs to be connected together to form a component-based system. Depending on the number of components stored, the number of possible combinations can be too high.

This way finding the most adequate component architecture to a given execution context will be a nontrivial job. This task is executed by the Optimizer component in our architecture; given an execution context sent by the client, the system will try to find a better suitable architecture by solving an instance of the component-oriented system optimization problem (COSOP).

3.1. Component-Oriented System Optimization Problem [13]. Any component-based system has at least one Provided Interface, that is used by external entities to access system's services and could has zero or more Required Interfaces that are used by the system to access external services.

In our case both types of interfaces are implemented by system's internal interconnected components. The interconnection is done by using the component's interfaces that could also be Required or Provided.

In Figure 5, the green and yellow circles represent system's provided and required interfaces. Dark blue and white circles represent components provided and required interfaces, respectively.

Each required interface represents a service needed by a component in order to work properly so it needs to be connected to a provided interface of another component to acquire the required service. This generates a compatibility property between the two interfaces that needs to be satisfied while connecting both.

This way a component-oriented system consists in interconnecting a set of components in such a way that the following rules are satisfied.

(i) All provided interfaces of the system are associated to a corresponding provided interface of an internal component.

(ii) All internal components' required interfaces are connected to a compatible component or associated to a corresponding system's required interface.

(iii) All provided interfaces could be connected to 0 or M_r required interfaces, where M_r is the maximum number of connections supported by the provided interface p.

Taking as an assumption that each component in REATIVO's repository can be evaluated by the Evaluator module, the cost of a component is defined as follows:

Cost evaluation function:

$$\text{cost}(c) = 100 - \text{evaluation}(c). \tag{1}$$

Therefore, using (1), the total cost of the system, represented by the sum of each system's component cost, must be minimized in order to achieve the best possible system.

3.2. Math Model. A mathematical model was developed in order to minimize the total system cost satisfying some constraints in order to maintain the properties of the component-based system. Based on the constraints to be satisfied and the minimization equation, we have the following optimization problem:

Equation to be minimized:

$$\text{minimize: } \sum_{i \in C} V_i \times X_i \tag{2}$$

within the following constraints.

Constraints of the minimization problem: Table 1 explains the complete set of variables considered in the optimization problem.

$$Y_{ikjl} - D_{kl} \le 0 \quad i \in C,\ k \in A_i,\ j \in C,\ l \in P_j, \tag{3a}$$

$$\left(\sum_{j \in C} \sum_{l \in P_i} Y_{ikjl} \times D_{kl} \right) - X_i = 0 \quad i \in C,\ k \in A_i, \tag{3b}$$

$$\left(\sum_{j \in C} \sum_{l \in P_i} Y_{ikjl} \times D_{kl} \right) - M_{jl} \times X_j \le 0 \quad j \in C,\ l \in P_j, \tag{3c}$$

$$X_i \in \{0, 1\}, \quad X_0 = 1 \quad i \in C, \tag{3d}$$

$$Y_{ikjl} \in \{0, 1\} \quad i \in C,\ k \in A_i,\ j \in C,\ l \in P_j. \tag{3e}$$

In this model, the objective function (2) has to be minimized in order to achieve the optimal component set to a given execution context.

The constraint (3a) means that only compatible interfaces could be connected, this way if a required interface k of a component i is connected to a provided interface l of a component j ($Y_{ikjl} = 1$) and the interfaces are not compatible ($D_{kl} = 0$), the constraint would be disrespected.

Table 1: Mathematic model.

Indexes	
i,j	Component indexes
k	Required interface index
L	Provided interface index
Parameters	
C	Set of components in the repository, C_0 is the system itself
A_i	Set of required interfaces of the component i
P_i	Set of the provided interfaces of the component i
M_{il}	Maximum connections supported by the interface l of the component i
V_i	Cost when using component i
D_{kl}	1 if the required interface k is compatible with the provided interface l
Decision variables	
X_i	1 if the component i was used in the current configuration; 0 otherwise
Y_{ikjl}	1 if the required interface k of the component i is connected in the provided interface l of component j; 0 otherwise

Restriction (3b) states that if a component was used in the solution, all its required interfaces must be connected to exactly one compatible provided interface. If one component i was used in the solution ($X_i = 1$), all its required interfaces k must be connected to exactly one compatible provided interface l of a component j, in this case, the equation results in 0.

However, (3c) indicates that the provided interfaces of a component j could only be used if it was used in the solution. In addition, the maximum number of connections in the provided interface l of component j has to be less than M_{jl}. If a provided interface l of a component j is used in the solution ($X_j = 1$), M_{jl} connections can be done between the component j and other components interfaces k that are compatible ($D_{kl} = 1$). This way, (3c) will result in 0 if the maximum number of connection is used or less than 0 if the number of connections established is less than M_{jl}.

3.3. Problem Complexity. The COSOP problem has a trivial solution if all components required interfaces are compatible with all other components provided interfaces, however, this is not the case for even the simplest system.

A better approach could be associate weights to a given interface connection. Compatible optimal interfaces have cost of 0, while incompatible interfaces have a high cost, other less suitable interfaces have intermediate costs, so the problem would be reduced to "interconnect a set of nodes (provided and required interfaces) through intermediate nodes (software components), minimizing the cost of interconnection." This problem was already solved and presented as the Stainer Tree Star [14, 15].

As the Stainer Tree Star problem is proved to be NP-Complete and our problem could be reduced to it, our problem can also be considered NP-Complete by the reduction proof.

3.4. Applying the GRASP Heuristic. As an NP-Complete problem, finding an optimal solution to a COSOP instance is not a trivial task. We decided to use the Greedy Randomized Adaptive Search Procedure (GRASP) [16] heuristic to find approximate solutions making the Optimizer component faster and reliable in terms of a real system.

The GRASP approach consists in an iterative algorithm where each iteration is composed of two stages: the build stage and a local search stage. The algorithm's final answer is the best solution among those found along successive iterations.

In the COSOP, the GRASP heuristic will consider C as the set of all components stored in the repository and M as the components that were assembled in the current solution. Two more sets are defined, A and P, where A is the set of the system and components' required interfaces in M that were not connected to provided interfaces and P the set of provided interfaces of the system and components' present in M.

In the build stage, it will be constructed a graph where the nodes represent the components and the edges represent the interconnections between their interfaces.

The evaluation cost and constraints to each solution are the same defined in (2) and (3a)–(3e), this way the algorithm will run until it finds a solution where all required interfaces of the components in M are connected to provided interfaces. In other words, the A set is empty.

3.5. Grasp Build Stage. This stage is initiated with an empty solution S in the graph. In each iteration, a required interface "a" from the set A and all components from C that has a provided interface compatible with "a" are chosen. The components are put in a Candidate List (CL) and then a Restricted Candidate List (RCL) is generated randomly using some of the best candidates in CL. In RLC will be components that satisfy (4) where $g(c)$ is the cost function given in (1).

Candidate evaluation:

$$g(c) \leq g_{\min} + \alpha(g_{\max} - g_{\min}). \tag{4}$$

After building the RCL, a component is chosen randomly from the set and the interface "a" that had been chosen in the beginning is connected to it. The Build Stage pseudocode is showed in Algorithm 1.

3.6. Local Search. In the local search stage, the neighborhood solutions of that one found in the build stage are generated using two perturbation operations: the SWAP operation, that consists in changing one component in the solution, and the DROP operation, that consists in removing components from the solution.

In both cases, using the SWAP or DROP operations implies in rebuilding a reliable solution by using the remaining components, however, those operations could improve the actual solution that is replaced by the better one in that case.

4. FlexTV Middleware Case Study

In this section, we analyse our case study, used in order to validate the REATIVO's approach. Current Digital TV (DTV) systems adopted around the world define middleware services in order to enable the TV receiver to execute TV application sent by the broadcasters in addition to the simple audio and video exhibition.

A DTV middleware basically standardize the application lifecycle and the APIs provided to the interactive applications running on top of it. In general, these middleware must deal with multiple context variations like different hardware platforms, support for broadcast applications, different signal quality and QoS requirements, and so forth. This way, a DTV middleware forms a relevant case study for the work proposed in this paper.

In this case study, we use a reference implementation of the Brazilian Digital TV procedural middleware called FlexTV. Our implementation follows a component-based architecture defining a set of loosed coupled components. Figure 6 presents a high level view of the FlexTV architecture showing the components and the execution environment.

The reception of low level streams is handled by the dark blue components. These components process the streams of audio, video and data, sending the video and audio to the *Media Processor* and the data streams to the *Data Processor*.

The yellow components are responsible to present media and user interface elements to the users and to capture the user interaction. For example, a key pressed in the user's remote control.

The applications that will execute over the middleware can use the *Application Communication* component to communicate with each other and the *Interaction Channel* component provides network access. The *Profile Manager* and *Persistence* are responsible for data storage. They store user preferences and other diverse data, respectively. The *Conditional Access* component handles access to restricted data and application security, authenticating applications that require privileged access to some system resources/functionalities. Finally, the *Application Manager* handles the load, configuration and execution of applications designed to run and use the middleware services.

The *Middleware Management* layer consists on the implementation of the REATIVO's client components described in Section 2.

As described in Section 2, the context variables are collected based on domain ontology. We developed a Digital TV ontology using the OWL language and the Protégé [17] tool. This Ontology contains 62 classes, 400 properties, and 33 relationships. The ontology classes are separated into three groups: classes related to user concepts, those related to the platform, and those related to the environment.

The user concepts are used to characterize the current users of the DTV system. The platform concepts are used to identify the current state of the DTV systems in terms of CPU

```
//A - list of system's access interfaces that are not connected
//P - List of system's provider interfaces
//C - List with the components that are in the repository
S = {} //solution graph empty in the beginning
while (A is not empty) do
{
        a = first(A)   //removes the first element from A
        if (exists (c)  in P where c has provider interfaces
                that is compatible with a){
        {
                connect a to its compatible interface in c
                continue
        }
        LC = {c in C where c has provider interfaces compatible with a)
        g_min = min { g(c) of LC }
        g_max = max{ g(c) of LC }
        LCR = { all c in LC where g(c) <= g_min + alpha * (g_max − g_min) }
        select a random member v in LCR
        S = S + {v}
        add the provider interfaces from v to P
        add the access interfaces from v to A
}
return S
```

Algorithm 1: The build stage.

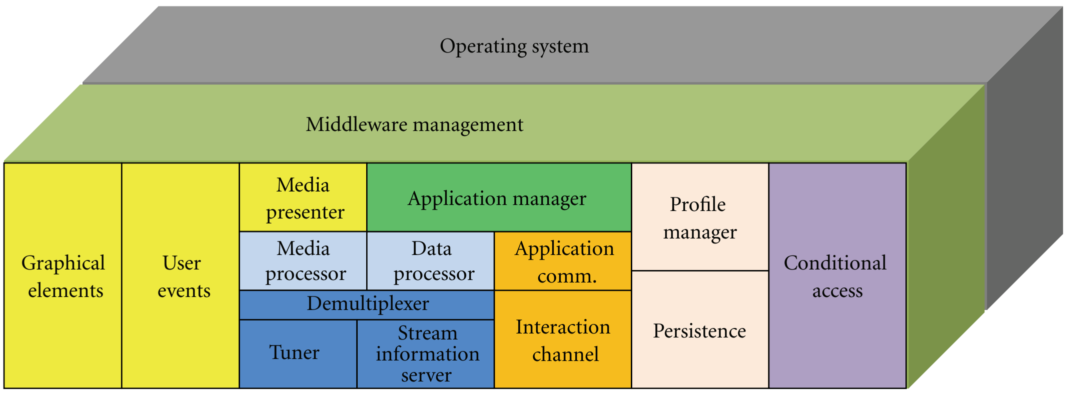


Figure 6: High level system architecture.

usage, memory usage, exhibition resolution, return channel type, and so forth. Finally, the environment variables can be used to define variables like the communication protocols currently being used, the network QoS, and so forth.

The concepts collected in this ontology were identified with 30 DTV researchers that worked in the Brazilian DTV system.

Based on this ontology, we can have a set of reconfiguration decisions. A very simple example can be shown in the case of the MediaProcessing component. Trojahn et al. [18] presents an evaluation of two Media Processing implementations. We can see that one of them requires less processing capabilities while the other one requires less memory. Based on this information and the information collected by the DTV client system, we can decide which one is more suitable for the moment. Of course, this example considers only two variables (processor and memory usage), but in the real case, several other variables may be considered as input to the Fuzzy logic engine.

5. Experimental Results

We performed a series of experiments with the middleware case study in order to get a clearer view of the feasibility of our approach.

As stated in the previous section, the client system components were developed in the Middleware Management layer of Figure 6. The tests were executed in a set-top box receiving the Digital TV signal through a 6 MHz broadcast channel and return channel based on PSTN modem. In order to execute the proof of concept, we generated a component descriptor with Fuzzy rules for two environment variables:

Table 2: Experimental results for GRASP based algorithms.

Instance		Optimal		GRASP			GRASPf		
ID	NC	z^*	T(s)	Z_{avg}	T(s)	Δ_{avg}%	Z_{avg}	T(s)	Δ_{avg}%
1	20	1184	1	1184	0.061	0.000	1184	0.409	0
2	30	592	2	600	0.031	1.351	592	0.376	0
3	50	870	30	927	0.06	6.552	870	0.44	0
4	60	609	109	650	0.067	6.732	613	0.449	0.657
5	80	—	—	1185	0.115	—	1106	0.729	—

device.resolution and *video.resolution*, referring, respectively, to the display device resolution and the received video resolution. After the fuzzy evaluation, the variables can be classified into three sets: *LDTV* (low definition), *SDTV* (standard definition), and *HDTV* (High definition). The output result can assume a value from 0 to 100 (zero means no adequacy, 100 means the best adequacy). We observed that the Fuzzy engine returned maximum adequacy level when the video resolution and the device resolution were equal. Minimum adequacy was received when we had big differences between video and display resolution.

In order to further validate our Fuzzy logic engine, 5 developers were invited to express their components adequacy in terms of fuzzy logic in order to feed the engine. Each generated component descriptor contained an average of 10 context variables. Accordingly to the developers, the values returned by the engine were consistent with their expected result.

Once the adequacy level is calculated, we can start the optimization stage. In this stage, the new architecture will be computed. As described in Section 4, the active repository uses a GRASP metaheuristic to compute the architecture and was developed using C++ language.

In order to test the GRASP algorithm, we generated several instances of the COSOP stated in Section 3.1 considering the number of components as NC = {20, 30, 50, 60, 80, 100, 500, 1000, 2000, 5000}. Each component in each instance was generated with a random number of required and provided interfaces. In these tests, the adequacy level was also randomly generated from the interval [50, 100].

We tested the instances with three algorithms: the GRASP, GRASP*f* (a variation of GRASP that chooses only the best solutions before entering the local search step), and Branch and Bound algorithms. The branch and bound algorithm is used in order to find the optimal solution, this way we are able to compare the GRASP-based algorithms with the optimal solution.

The results can be found in Table 2.

The z^* indicates the optimal solution found by the Branch and Bound method. $T(s)$ indicates the time, in seconds, spent by the approach to reach the solution. The Z_{avg} column indicates the average of the solutions found after five executions of the metaheuristics. Finally, the following equation is used to evaluate the average of the delta value that indicates the difference between the Z_{avg} and the optimal solution.

Average of the delta value:

$$\Delta_{\text{AVG}} = \left[\frac{(Z_{\text{AVG}} - z^*)}{z^*} \right] \times 100. \tag{5}$$

According to Table 2, the GRASPf algorithm found the best solution when NC equals to 20, 30 and 50. In the instance number 4 (NC = 60) the gap was less than 1%. On the other hand, the GRASP algorithm results had gaps of more than 1%.

With respect to computation time, we can see that the time to execute a simple instance with 60 components were more than 1 minute in the case of the optimal solution. Furthermore, the optimal solution could not compute the instance 5 (NC = 80) because there was no system resources to instantiate the problem. The branch and bound results were obtained using the LINGO Optimization Modeling software [19].

In order to evaluate our approach with the case study, we employed our active repository to receive context information from a set of FlexTV middleware instances. In this case, the context information was sent from the middleware to the repository as an OWL instance file. This OWL representation was generated using the Jena framework [20]. This framework provides, among other tools, an API to read and modify OWL models. Through this API, we were able to verify which concepts are present in the ontology and instantiate these concepts associating values to their properties. Once the context is delivered to the active repository, the components start to be evaluated accordingly to it. The Fuzzy engine was implemented as a Fuzzy inference system using Matlab [21] mathematical programming tool. The Optimizer is responsible to calculate the new component architecture using algorithms discussed in Section 3. Generating optimized component architectures generates adapted architectures and reconfiguration commands and sends to the client system by using a XML file.

In the user point of view, the time spent by the system to perform the necessary data collection, to send the context information to the server and to receive and load components into memory are irrelevant since these operations occur in background, so no impact to the user experience is noted. Only the time spent during the system reconfiguration is relevant, since during this period the system will be unavailable to the user. In the performed tests, we verified that the maximum time that the system became unavailable to the user was 2,190 ms, in this specific case, most of the time was spent changing the components responsible to handle the video stream (1,800 ms). The restart process of these components is slow and similar to the time required to perform a channel change in the digital TV receiver.

Considering that no reconfigurations caused failures in the system and that the worst unavailable time was similar to the channel change time (which the users are already used to), the test results are considered to be satisfactory. Furthermore, if we decide to change some component that is not related to the video stream chain, the video will not be disrupted and the reconfiguration may be totally seamless to the user.

6. Related Works

Component based software is an area that have grown in the pass years, the concept of components and their capacity of improve the software development cycle have created a vast number of architectures to support component-based systems.

6.1. Code Conjurer. Code Conjurer [22] works as a component finder that analyses the developer code in project or writing stages. To do the component search it uses the component repository Merobase [23]. Due to the high number of components found every time in Merobase's search, many components are high reclined to fail in the functionality aspects. To overcome this issue the Code Conjurer implements a test-driven search that will test the components found by using user-defined test cases to the system and then suggest only that ones that passed all test cases. This way the developer will have a better quality of the suggested components. Furthermore, the developer can also employ UML diagrams and descriptions to suggest eligible components.

There are several differences between our work and that presented in Code Conjurer. First the information used by it is focused in the programmer's context, our system uses contextual information that comes from the execution context. Second, Code Conjurer indexes source code that the programmer can adapt to his system in development time, the components stored in our system are already compiled within a well-defined component model.

There are many other software that use components as the Code Conjurer tool [11, 24] and others that follow a similar category but focus on helping the developers navigating the vast number of APIs provided by their currently used languages [25, 26], however, as mentioned before they focus on accelerating code development not focusing on running systems.

6.2. Fractal. Fractal [27–29] is a hierarchical and reflective component model that provides functionalities such as inspection and change of component's internal behavior. In order to change the components internal behavior, Fractal defines a flexible component system. The main concept is that Fractal components work as a membrane for existent components that allows introspection, composition, sharing, and reconfiguration capabilities.

A Fractal component is composed of a membrane and its contents. The content is a finite set of components. The membrane is an encapsulation layer that can have external interfaces, which can be accessed by external components; internal interfaces, which can be accessed only by internal component; controllers, which are used to, among other functions, inspect the component's internal contents, control internal components behavior and intercept actions.

To test the proposed model the authors have implemented Julia, a java version of the component model that implements many of the Fractal's capabilities. The results are presented as an evaluation of Julia's implementation in terms of overhead, memory allocation, and framework usage. The main difference between Fractal's architecture and the architecture considered in our approach is that Fractal the first one focus on software components and forms of handling them in different ways. A possible future work is to evaluate Fractal as the component model used by the middleware implementation. This way, we can identify possible positive and negative aspects and extension points to Fractal.

6.3. OpenCOM. OpenCOM [30] is a component-based technology that is programming language independent. OpenCOM was designed to support the development of general systems while allowing run time configuration.

In the OpenCOM model, components are loaded in a macro component called Capsule through a set of APIs called Component Runtime Kernel (CRTK) containing all configuration operations (such as load, connect, disconnect, among others) associated to a rollback mechanism that is used in case of failure. Components are functional units that are encapsulated and have Receptacles and Interfaces that are, respectively, required and provided.

OpenCOM support additional services that are implemented as components thus nothing changes in the component model, like the reflexive meta-models that improves the dynamic reconfiguration of systems. Among the reflexive meta-models those that are more related with our work are what follows.

(i) Architectural meta-model: exposes the architectural composition of the components in a Capsule as a connected graph.

(ii) Interface meta-model: Allows the discovering of information about the component's interface types and the using of dynamic interface discovery.

(iii) Interception meta-model: Allows the creation of interceptors between interfaces and receptacles.

(iv) Resource meta-model: Offers the access to the low level platform resources such as threads, procedures, memory, and others.

As in the case of Fractal, we also identify as future work, the possibility to evaluate OpenCOM in REATIVO's context.

6.4. Design and Implementation of a Safe, Reflective Middleware Framework. This work presents a model for composition of safe middleware services where applications could have variant nonfunctional requirements over the time [31].

To make this possible, the authors identify core services that are the basis to other services in order to make possible the variation of the requirements. According to the authors, this model handles the complexity of reasoning over the components in distributed system in order to provide a better component if such is available.

The new composed services have to be weakly coupled with the application requirements so these components could be changed over the time without any major reconfiguration issue. In our work, we go further by providing a complete infrastructure for context evaluation and architecture reconfiguration.

7. Conclusion and Future Directions

This paper presented an approach to adapt context-aware component-based software systems. A new active repository approach is employed in which components are made available not only during software development phase, but also during the operational phase of the software in a context aware form. In this context, the repository is able to receive the client system context and generate a new architecture that better fit this context.

We defined the concept of execution context as an instance of a domain dependent ontology. This ontology defines a set of variables that are relevant to the system in terms of functionality.

In order to decide which components are more suitable to a given context, we employed a fuzzy logic approach to calculate the adequacy level of the components. To be compatible with REATIVO, components registered in the repository must provide a set of definitions as parameters to the fuzzy inference engine.

REATIVO is able to evaluate the entire system adequacy by assembling the components in a system composed by the "most adequate" components. The Component-Oriented Software Optimization Problem is proven to be NP-Complete, thus it would need an optimization approach. In this context, we employed GRASP metaheuristic to compute the optimized architecture.

In order to achieve a more concrete result, we applied our solution to the FlexTV Middleware case study and obtained some convincing results regarding the feasibility of the approach, since both the client side and server side were successfully implemented and the time spent in reconfiguration procedures were satisfactory.

In Section 2.2, we stated that REATIVO's repository is domain independent. As a future work, we need to employ REATIVO in new domains other than digital TV in order to provide empirical proof of this statement, in addition to the reasons presented in that section. Furthermore, the application of different algorithms to substitute the fuzzy logic and GRASP is also desired. In this way, a better evaluation can be performed related to the efficiency of the chosen algorithms.

References

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[2] C. Szyperski, D. Gruntz, and S. Murer, *Component Software—Beyond Object-Oriented Programming*, ACM Press, 2nd edition, 2002.

[3] G. T. Heineman and W. T. Councill, *Component-Based Software Engineering: Putting the Pieces Together*, Addison-Wesley, 2001.

[4] C. W. Krueger, "Software reuse," *ACM Computing Surveys*, vol. 24, no. 2, pp. 131–183, 1992.

[5] Component Source, 2012, http://www.componentsource.com.

[6] S. Henninger, "Using iterative refinement to find reusable software," *IEEE Software*, vol. 11, no. 5, pp. 48–59, 1994.

[7] K. Inoue, R. Yokomori, H. Fujiwara, T. Tamamoto, M. Matsushita, and S. Kusumoto, "Component rank: relative significance rank for software component search," in *Proceedings of the 25th International Conference on Software Engineering (ICSE '03)*, pp. 14–24, Portland, Ore, USA, May 2003.

[8] C. J. M. Geisterfer and S. Ghosh, "Software component specification: a study in perspective of component selection and reuse," in *Proceedings of the 5th International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems*, p. 9, February 2006.

[9] B. Srivastava, K. Ponnalagu, N. C. Narendra, and K. Kannan, "Enhancing asset search and retrieval in a services repository using consumption contexts," in *Proceedings of IEEE International Conference on Services Computing (SCC '07)*, pp. 316–323, July 2007.

[10] D. Lucrécio, A. Prado, and E. Santana, "A Survey on software components search and retrieval," in *Proceedings of the 30th Euromicro Conference (Eromicro '04)*, pp. 152–159, 2004.

[11] Y. Ye, *Supporting component-based software development with active component repository systems [Ph.D. thesis]*, Department of Computer Science, University of Colorado, 2001.

[12] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.

[13] L. Leite, G. Farias, G. Souza, and S. Meira, "Uma Meta-heurística GRASP para Otimização de Sistemas Orientados a Componentes de Software," in *Anais do Simpósio da Sociedade Basileira de Pesquisa Operacional*, Ceará, Brazil, 2007.

[14] P. Winter and M. Zachariasen, "Euclidean Steiner minimum trees: an improved exact algorithm," *Networks*, vol. 30, no. 3, pp. 149–166, 1997.

[15] Y. Lee, S. Y. Chiu, and J. Ryan, "A branch and cut algorithm for a Steiner tree-star problem," *INFORMS Journal on Computing*, vol. 8, no. 3, pp. 194–201, 1996.

[16] M. Resende, "Greedy randomized adaptive search procedures (GRASP)," in *Encyclopedia of Optimization*, vol. 2, pp. 373–382, Kluwer Academic Publisher, 2001.

[17] Stanford Center for Biomedical Informatics Research, "The Protégé Ontology Editor and Knowledge Acquisition System," 2012, http://protege.stanford.edu/.

[18] T. H. Trojahn, J. L. Goncalves, J. C. B. Mattos et al., "Tests and performance analysis of media processing implementations for the middleware of Brazilian Digital TV system using different scenarios," in *Proceedings of the 5fth International Conference on Multimedia and Ubiquitous Engineering (MUE '11)*, pp. 95–100, June 2011.

[19] Lindo Systems, "Lingo—Optimization Modeling Software for Linear, non-linear and Integer Programming," 2012, http://www.lindo.com/index.php?option=com_content&view=article&id=2&Itemid=10.

[20] The Apache Software Foundation, "Apache Jena," 2012, http://incubator.apache.org/jena/.

[21] MathWorks, "Matlab—The Language of Technical Computing," 2011, http://www.mathworks.com/products/matlab/.

[22] O. Hummel, W. Janjic, and C. Atkinson, "Code conjurer: pulling reusable software out of the thin air," *IEEE Software*, vol. 25, no. 5, pp. 45–52, 2008.

[23] "Merobase—Software Component Finder," 2012, http://www.merobase.com.

[24] F. McCarey, M. Ó. Cinnéide, and N. Kushmerick, "Rascal: a recommender agent for agile reuse," *Artificial Intelligence Review*, vol. 24, no. 3-4, pp. 253–276, 2005.

[25] D. Mandelin, L. Xu, R. Bodík, and D. Kimelman, "Jungloid mining: helping to navigate the API jungle," in *Proceedings of the ACM SIGPLAN Conference on Programming Language*

Design and Implementation (PLDI '05), pp. 48–61, ACM Press, Chicago, Ill, USA, June 2005.

[26] S. Thummalapenta and T. Xie, "Parseweb: a programmer assistant for reusing open source code on the web," in *Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE '07)*, pp. 204–213, ACM Press, New York, NY, USA, November 2007.

[27] G. Blair, T. Coupaye, and J.-B. Stefani, "Component-based architecture: the Fractal initiative," *Annals of Telecommunications*, vol. 64, no. 1-2, pp. 1–4, 2009.

[28] P. Merle and J.-B. Stefani, "A formal specification of the fractal component model in alloy," INRIA Research Report 6721, 2008.

[29] E. Bruneton, T. Coupaye, M. Leclercq, V. Quéma, and J.-B. Stefani, "The FRACTAL component model and its support in Java," *Software—Practice and Experience*, vol. 36, no. 11-12, pp. 1257–1284, 2006.

[30] G. Coulson, G. Blair, P. Grace et al., "A generic component model for building systemssoftware," *ACM Transactions on Computer Systems*, vol. 26, no. 1, article 1, 2008.

[31] N. Venkatasubramanian, S. Gutierrez-nolasco, S. Mohapatra et al., "Design and Implementation of a Safe, Reflective Middleware Framework," CiteSeerX—Scientific Literature Digital Library and Search Engine, 2008.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 748163, 17 pages
doi:10.1155/2012/748163

Research Article

Exploiting Location and Contextual Information to Develop a Comprehensive Framework for Proactive Handover in Heterogeneous Environments

G. Mapp,[1] F. Katsriku,[2] M. Aiash,[1] N. Chinnam,[1] R. Lopes,[3] E. Moreira,[3] R. M. Porto Vanni,[4] and M. Augusto[5]

[1] *School of Engineering and Information Sciences, Middlesex University, Hendon, London NW4 4BT, UK*
[2] *Department of Computer Science, University of Ghana, Accra, Ghana*
[3] *Mathematics and Computer Sciences Institute, University of Sao Paulo, 13566-590 Sao Carlos, SP, Brazil*
[4] *Department of Informatics, Federal Institute of Education, Science and Technology of Sao Paulo, 14801-600 Araraquara, SP, Brazil*
[5] *Department of Information Systems, Santa Catarina State University, 89283-081 Sao Bento do Sul, SC, Brazil*

Correspondence should be addressed to G. Mapp, g.mapp@mdx.ac.uk

Received 10 March 2012; Revised 7 June 2012; Accepted 8 June 2012

Academic Editor: Hoon Ko

The development and deployment of several wireless and cellular networks mean that users will demand to be always connected as they move around. Mobile nodes will therefore have several interfaces and connections will be seamlessly switched among available networks using vertical handover techniques. Proactive handover mechanisms can be combined with the deployment of a number of location-based systems that provide location information to a very high degree of accuracy in different contexts. Furthermore, this new environment will also allow contextual information such as user profiles as well as the availability of using location and contextual information to provide efficient handover mechanisms. Using location-based techniques, it is possible to demonstrate that the Time Before Vertical Handover as well as the Network Dwell Time can be accurately estimated. These techniques are dependent on accurately estimating the handover radius. This paper investigates how location and context awareness can be used to estimate the best handover radius. The paper also explores how such techniques may be integrated into the Y-Comm architecture which is being used to explore the development of future mobile networks. Finally, the paper highlights the use of ontological techniques as a mechanism for specifying and prototyping such systems.

1. Introduction

The development and deployment of several wireless networks mean that mobile devices will have several wireless interfaces including 3G, WLAN, WiMAX, and LTE. This represents a significant development as users will want to be always connected from anywhere and at any time. This will be achieved using vertical handover techniques where connections will be seamlessly switched between available networks. As explained in [1], handover is defined as the changing of the network point of attachment (POA) of a mobile device. When the device moves to a new point of attachment which is technologically identical to the previous point of attachment, this is called horizontal handover. Vertical handover is defined as a handover where the new point of attachment comprises a different technology when compared with the previous point of attachment [2].

Proactive handover in which the mobile node actively attempts to decide when and where to handover can help to develop an efficient handover policy mechanism. This is because by using proactive handover, it is possible to minimize packet loss and service disruption as an impending handover can be signalled to the higher layers of the network protocol stack. However, in order to develop efficient proactive handover, location information and services are necessary. In addition, context-aware information based on user profiles, local network services, and information from other users can be used to ensure that the mobile user is *always best connected* or (ABC) [3].

Information such as the position of Access Points, network coverage as well as the speed and direction of the mobile node can be used to develop equations which can calculate *Time Before Vertical Handover* or TBVH as well as the estimated *Network Dwell Time* or NDT in a given network. This paper develops a mathematical framework which can be used to calculate TBVH and NDT. Knowing TBVH and NDT allows mobile nodes to efficiently decide when is the best place to do a handover, such that there is very little handover latency or packet loss. It will be shown that this technique is dependent on having a good estimation of the handover radius which is, in turn, dependent on an accurate propagation model that needs to be dynamic and adaptable. This paper shows how contextual information about network conditions encountered by other mobile nodes could be used to build a more dynamic and adaptable propagation model.

Since it is not possible for mobile nodes to store the information on all the available networks, it is necessary that such information be available in the core network. This could be implemented as part of core network functionality for future mobile networks. Such an approach is being adopted as seen in the Media-Independent Information Service (MIIS) advocated by the IEEE 802.21 standards [4] which look at specifying Media Independent Handovers (MIH). However, this paper argues that in order to build a comprehensive platform, it is necessary to include information on users as well as on networking infrastructure.

This paper attempts to bring together several aspects of handover to form a coherent framework. It shows how these methods can be used to provide mobile nodes with effective handover policies and how such policies can be implemented on an architecture such as Y-Comm which is being used to build future mobile systems.

The unique contributions of this paper can be described as follows: using location information and an estimation of the handover radius, this work shows how TBVH and NDT are calculated. It shows how Media Independent Information Service (MIIS) can be extended to store location information on networking infrastructure. This work also shows how contextual information about the mobile node in terms of the signal strength needed to handover, as well as information about its previous journeys and other mobile nodes can be combined using Wireless Footprinting techniques. This contextual information can also be used to develop a more dynamic and adaptable propagation model. Finally, this paper shows how an ontology can be used as an implementation path to develop a prototype testbed. Compared to previous studies, this work presents a clear way forward for the development and deployment of proactive mechanisms in real mobile systems.

The rest of the paper is structured as follows: Section 2 discusses handover classification while Section 3 examines how handover coverage is specified. In Section 4 the mathematical framework is detailed while Section 5 examines the application of the mathematical framework to a specific network scenario. Section 6 looks at results for different handover radii. Section 7 investigates how the information on local networks can be acquired and stored. Section 8 looks at the Y-Comm Framework, particularly how Y-Comm deals with handover. Section 9 summarises previous work while the paper concludes with Section 10 which looks at conclusions and future work.

2. Vertical Handover: A Detailed Classification

In this section, we define the set of terms used in the context of handover. This presentation is divided into two broad types. The first has to do with a general handover classification which classifies handovers using general characteristics while the second is a more advanced classification dealing with the different mechanisms and inputs to do with handover.

2.1. General Handover Classification. Handovers can be classified in terms of when the connection to the previous POA is broken in relation to when the connection to the next POA is made, which device is in charge of handover, and the coverage areas of the relevant networks.

(i) Hard versus soft handover: in a *hard handover*, the connection to the previous POA is broken before the connection to the new POA is made. In *soft handover*, the connection to the next POA is made before the connection to the previous POA is broken.

(ii) Network-based versus Client-Based Handover: in *network-based handover*, the network makes the final handover decision while in *client-based handover*, it is the mobile device that decides when to handover. Though current cellular systems use network-based handovers, mechanisms such as Mobile IP [5] and Fast Mobile IP [6] are client-based. In fact, for heterogeneous environments client-based handover is favoured [7].

(iii) Upward versus downward Handover: where a network, say A, is completely covered by network B, then if we make a handover from network A to network B, this is referred to as an *upward handover* because we are going from a smaller network with substantial bandwidth to a network of a much larger coverage with lower bandwidth. While a handover from network B to network A is referred to as a *downward handover* because we are going from a larger to a smaller network.

2.2. Advanced Classification. An advanced type of handover classification is shown in Figure 1. Handovers can also be divided into two advanced types. Imperative handovers occur due to technological reasons only. Hence the mobile node changes its network attachment because it has determined by technical analysis that it is good to do so. This could be based on parameters such as signal strength, coverage, and the quality-of-service offered by the new network. These handovers are imperative because there may be a severe

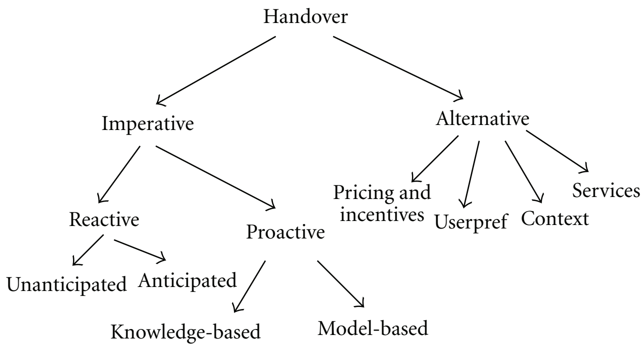


Figure 1: Handover classification.

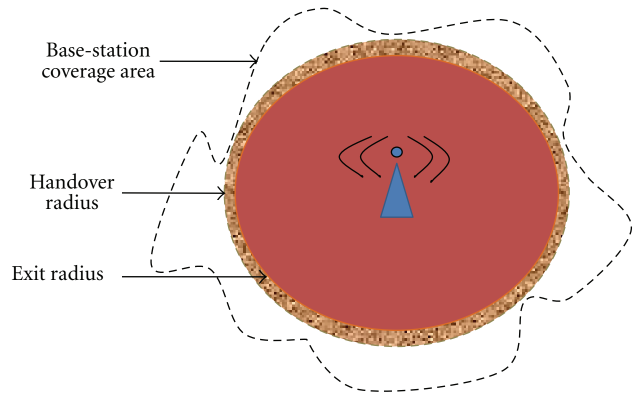


Figure 2: Network coverage.

loss of performance or loss of connection if they are not performed. In contrast, alternative handovers occur due to reasons other than technical issues [8]. Hence there is no severe loss of performance or loss of connection if an alternative handover does not occur. The factors for performing an alternative handover include a preference for a given network based on price or incentives. User preferences based on features or promotions as well as contextual issues might also cause handover. Finally, there may be other network services that are being offered by certain networks.

Imperative handovers are, in turn, divided into two types. The first is called *reactive handover*. This responds to changes in the low-level wireless interfaces as to the availability or nonavailability of certain networks. Reactive handovers can be further divided into anticipated and unanticipated handovers [9]. Anticipated handovers are soft handovers which describe the situation where there are alternative base-stations to which the mobile node may handover. With unanticipated handover, the mobile node is heading out of range of the current POA and there is no other base-station to which to handover. These handovers are therefore examples of hard handovers.

The other type of imperative handover is called *proactive handover*. These handovers use soft handover techniques. Proactive handover policies attempt to know the condition of the various networks at a specific location before the mobile node reaches that location. Proactive policies allow mobile nodes to calculate the *Time Before Vertical Handover* (TBVH) which enables them to minimize packet loss and latency experienced during handovers. Presently, two types of proactive handovers are being developed. The first is knowledge-based and attempts to know, by measuring beforehand, the signal strengths of available wireless networks over a given area such as a city. This could involve physically driving around and taking these readings [10]. These measurements need to be taken in different seasonal contexts as the effects of foliage on wireless propagation are well known [11]. The second proactive policy is based on a mathematical model which calculates the point when vertical handover should occur and the time that the mobile would take to reach that point based on its velocity and direction [12]. The accuracy of this approach is dependent on various factors including location technology, the propagation model used, network topology, and specific environments, for example, whether the mobile node is indoor or outdoor [13]. In this paper, we will develop algorithms based on the model-based approach.

3. Handover Coverage Parameters

In this section, we introduce a set of network coverage terms. We first define the concept of a coverage area as the region where signals from a given POA can be detected. The coverage area is of an irregular shape. On the boundary of the coverage area signals from the POA are unreliable and beyond the coverage area, the signals from the POA cannot be detected. Since handover is a complex operation, it should be finished before the coverage boundary is reached. So we define a circle known as the *handover threshold*, and the corresponding radius is called the *handover radius* represented by R_H. Ideally, it is necessary to complete handover before or at the handover boundary to ensure smooth handover. In order to make sure that handover occurs at the handover boundary, we define another coverage area known as *exit coverage* represented by the *exit radius* or R_E. Handover must begin at the exit radius in order to ensure that it is completed at the handover boundary. The arrangement is shown in Figure 2 and discussed in [14]. The exit radius will therefore be dependent on the velocity, vel, of the mobile node. If we represent the time taken to execute a handover by T_{EH}, then we can say in

$$T_{EH} \leq \frac{(R_H - R_E)}{\text{vel}}. \tag{1}$$

Hence

$$R_E \leq R_H - (\text{vel} * T_{EH}). \tag{2}$$

So the faster a mobile node moves the smaller the exit radius at which handover must begin. We will now continue our analysis using the handover radius. But given that we can find the handover radius and we know the velocity of the mobile node and the time taken to execute a handover, then we can calculate the exit radius.

3.1. Estimating Handover Radius Using Propagation Models. The accuracy of the system revealed above depends on a good estimation of the handover radius which, in turn, is dependent on the propagation models being used. Propagation models attempt to model the received signal strength of an electromagnetic signal at a given distance from where the signal is being transmitted. Propagation models are divided into 3 main types.

(i) *Empirical Models*. These models are based on mathematical equations that attempt to take into account the travelling effects on the electromagnetic signal [15]. Empirical models can further be divided into two categories: time dispersive and nontime dispersive models. Time dispersive models take into account channel characteristics such as multipath spread. Nontime dispersive models predict the path loss in terms of distance, height of antenna, and frequency. These models are mainly based on observations as well as measurements. Hata and Cost-231 are examples of nontime dispersive models.

(ii) *Deterministic Models*. These models are related to the propagation of electromagnetic waves to find out the strength of the received signal at the particular point. It requires a 3D view of electromagnetic waves. Ray tracing is an example of deterministic models.

(iii) *Stochastic Models*. These models generate predictions with the help of a series of random variables; these models are therefore less accurate than predictions based on empirical or deterministic models. But stochastic models require less information and have lower processing power requirements.

This paper concentrates on nontime dispersive empirical models.

3.2. Path Loss Models. In wireless communications there is some loss of signal strength between the transmitter and receiver. This is called *path loss* and is the essential parameter used to predict the strength of signals in radio systems at various locations. The path loss is also dependent on environmental conditions and will be different in rural, urban, and suburban environments. Path loss models are usually dependent on both the distance from the transmitter as well as the frequency of the signal being transmitted. We can calculate the path loss in all these environments with the help of path loss models. Path loss can therefore be used to find the handover radius.

3.2.1. Free Space Models. This model deals with path loss in free space. In this environment, the attenuation is proportional to the square of the distance between the transmitter and the receiver and also to the square of the frequency of the radio signal.

The Free Space Path Loss (FSPL) in decibels is given by

$$\mathrm{FSPL} = 20\log_{10}(d) + 20\log_{10}(f) + 32.44 - G_t - G_r, \quad (3)$$

where d is the distance from the antenna in kilometres, f is the frequency of the signal in MHz, G_t is the gain at the transmitter, and G_r is the gain at the receiver.

This formula is effective where multipath effects are minimal such as for satellites in space.

3.2.2. The Okumura Model. In urban and suburban areas, it is necessary to consider multipath effects. The Okumura model is used to model urban areas. The path loss (PL) is given by

$$\mathrm{PL} = L_f + A_{\mathrm{mu}} - G_{\mathrm{ht}} - G_{\mathrm{hr}} - G_{\mathrm{area}}, \quad (4)$$

where L_f is the free propagation path loss, A_{mu} is the attenuation of the medium relative to free space, G_{ht} is the gain due to the height of the base station antenna, G_{hr} is the gain due to height of the mobile node receiver, and G_{area} is the gain due to the type of environment.

3.2.3. Hata Models. The Okumura model has been used as a base to evolve Hata models which can be deployed for urban and suburban environments.

For the urban areas,

$$\begin{aligned} L_u = {} & 69.55 + 26.16\log_{10}f - 13.82\log_{10}h_b - C_H \\ & + \left(44.9 - 6.55\log_{10}h_b\right)\log_{10}d, \end{aligned} \quad (5)$$

where h_b is the height of the base station, h_m is the height of the mobile node, and C_H is the antenna height correction factor.

For small or medium cities,

$$C_H = 0.8 + \left(1.1\log_{10}f - 0.7\right)h_m - 1.56\log_{10}f. \quad (6)$$

For large cities, the formula is dependent on two frequency ranges.

So for $150 \leq f \leq 200$ Mhz

$$C_H = 8.29\left(\log_{10}(1.54h_m)\right)^2 - 1.1 \quad (7)$$

and for 200 MHz $\leq f \leq$ 1500 Mhz

$$C_H = 3.2\left(\log_{10}(11.75h_m)\right)^2 - 4.97. \quad (8)$$

For Suburban areas,

$$L_{su} = L_u - 2\left(\log_{10}\left(\frac{f}{28}\right)\right)^2 - 5.4, \quad (9)$$

where L_u is the path loss in urban areas, f is the transmission frequency in MHz, and all path loss values are in dB.

3.3. Using the Propagation Models. In order to find out the best handover radius we need the propagation characteristics as well as two other key parameters. The first is the power with which the signal is being transmitted. Hence the power of the signal at the transmitter is important as a more powerful signal would mean that a large handover radius is possible. The second factor is the signal strength threshold that is used to handover. Here, an absolute signal threshold could be used or a relative threshold could also be employed.

Most Wireless LAN NIC cards can work between −70 and −85 dB [16]. For WLANs, it has been decided to use the Free Space Model due to the high frequency being used. For cellular networks, Okumura and Hata models perform better than Free Space Models.

3.3.1. Network Interaction Classification. In order to build a complete framework, it is not only necessary to consider individual networks but also how individual networks in a given vicinity will interact with each other. This is even more important as there are completely different policies that are used for the deployment of different kinds of wireless networks. Cellular networks are rolled out on a national scale and issues of network coverage and user access are very carefully studied so that the deployment of masts and base-stations will be done in a holistic way. In contrast, WiFi networks tend to be set up individually and in an ad-hoc manner. This means that we can consider the interaction of these networks in general. We do this by looking at the relationship between Network A and Network B. There are 3 cases to consider.

(i) The first is that Network A is completely covered by Network B. This occurs, for example, between a single local wireless LAN network and a colocated cellular network. In this context, mobile nodes moving in network B and if, depending on their movement, they come into the range of network A, then they would do an alternative downward handover to network A. After travelling through Network A, the mobile node would then do an imperative upward handover back to Network B.

(ii) The second network relationship occurs when Network A and Network B intersect. In this context, we need to understand the area of joint coverage and the implications for handover policy. If, given the mobile's direction, we can work out the distance for which the mobile node will be in contact with both networks, then we can work out the earliest and latest times at which a handover can occur.

(iii) The third network relationship is that Network A and Network B do not overlap at all. Hence the mobile node travelling between the two networks must do a hard handover when moving from Network A. It would shut down its connections, move to network B where it must contend for a channel, and then it would handover to the new channel and reopen its connections [17].

4. The Mathematical Framework

The Law of Cosines. In order to build a mathematical framework we need mathematical tools. An important tool is the *law of cosines*. The laws are given by the following equations.

$$c^2 = a^2 + b^2 - 2ab\cos\gamma,$$
$$a^2 = b^2 + c^2 - 2bc\cos\alpha, \tag{10}$$
$$b^2 = a^2 + c^2 - 2ac\cos\beta.$$

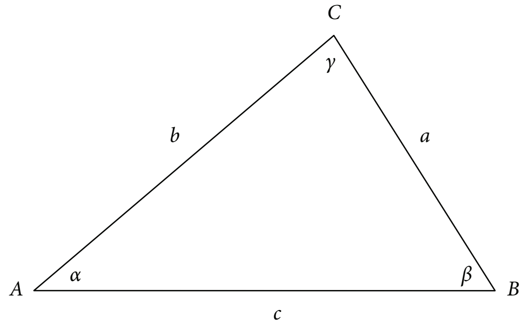


Figure 3: Law of cosines.

The sides and angles are shown in Figure 3. Further formulas can be derived, namely:

(i) the third side of a triangle if one knows two sides and the angle between them:

$$c = \sqrt{(a^2 + b^2 - 2ab\cos\gamma)}, \tag{11}$$

(ii) the angles of a triangle if one knows the three sides:

$$\gamma = \arccos\left(\frac{(a^2 + b^2 - c^2)}{2ab}\right), \tag{12}$$

(iii) the third side of a triangle if one knows two sides and an angle opposite to one of them:

$$a = b\cos\gamma \pm \sqrt{\left(c^2 - b^2\sin^2\gamma\right)}. \tag{13}$$

4.1. Calculation for Different Handover Scenarios. In the section we look at different handover scenarios and calculate the Time Before Vertical Handover (TBVH) distance as well as the Network Dwell (ND) distance where appropriate. We now denote the handover radius by R.

4.1.1. Complete Coverage. For complete coverage we look at when Network A is completely covered by Network B, so Network A is a WLAN network and Network B is a cellular network such as GSM or UMTS. We now investigate upward handover as shown in Figure 4.

In this figure, A is the centre of the WLAN, the mobile node is located at C, and knows its location using a GPS device. $R1$ is the handover radius. The line CF represents the present trajectory of mobile node. Since we know $d1$, $R1$, and angle x, as shown in [13], we can use (13) to obtain CF as:

$$CF = d1\cos x \pm \sqrt{\left(R1^2 - d1^2\sin^2 x\right)}. \tag{14}$$

Hence TBVH = CF/vel.

Next we consider downward handover, where the mobile node is going from the larger coverage network to a smaller network. This is shown in Figure 5.

The WLAN network is given by its radius $R2$ and centre B. The mobile node is currently located at C and is travelling

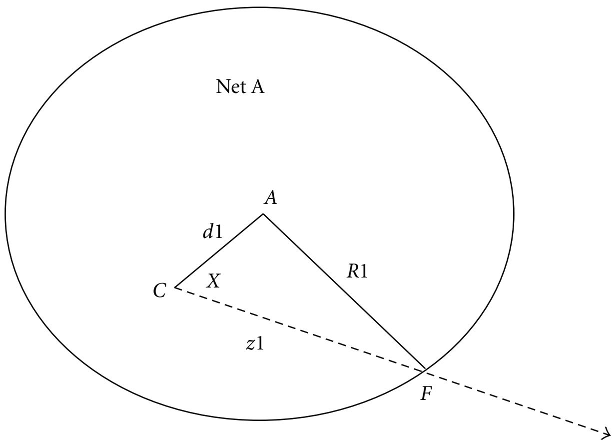


Figure 4: Upward handover.

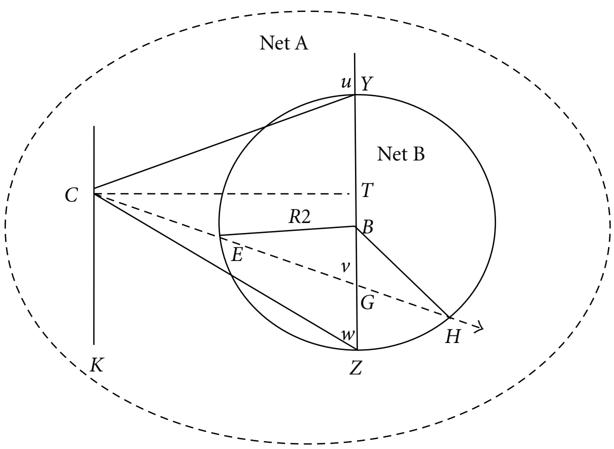


Figure 5: Downward handover.

on trajectory $CEGH$. The vertical axis of the WLAN is given by the points Y and Z. These points can be easily obtained by the mobile node because it knows $R2$ and B. From knowing Y and Z it is possible to calculate CY and CZ, respectively.

In terms of angles, we define them as relative to the vertical axis YZ, so CY makes the angle u with YZ while CG makes the angle v and CZ makes the angle w with YZ. We can measure those angles directly from C since the angle u is YCK, the angle v is GCK, and the angle w is ZCK. So the first thing we need to find out is if the trajectory of the mobile node means that it will or will not encounter the smaller network. We can easily verify this using angles u, v, and w. So we know that the mobile node will hit the WLAN because $u \geq v \geq w$. If $v > u$ or $v < w$ then the mobile node will not encounter the WLAN at all.

Since we can measure these angles and distances, we can now calculate CG. We use CY since

$$CT = CY \sin(180 - u) = CG \sin v,$$
$$CG = \frac{(CY \sin(180 - u))}{\sin v}. \tag{15}$$

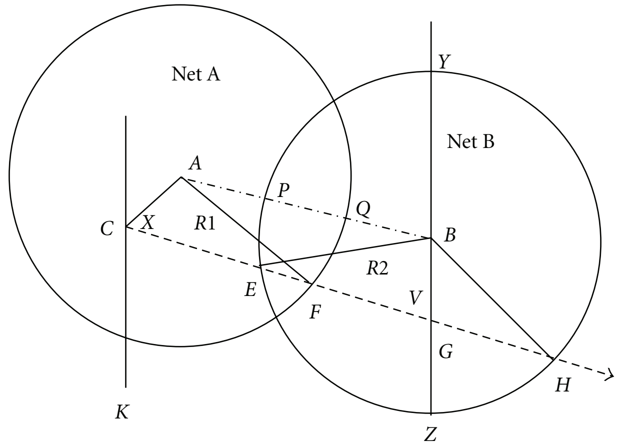


Figure 6: Intersecting networks.

So we know CZ, CG and CGZ is $(180 - v)$; we can get GY from (13)

$$GZ = CG \cos(180 - v) \pm \sqrt{\left(CZ^2 - CG^2 \sin^2(180 - v)\right)}. \tag{16}$$

Since $BG = BZ - GZ$ and $BZ = R2$ then $BG = R2 - GZ$. So we can calculate BG. We know $BE = R2$, BG, and angle v; hence

$$EG = BG \cos v \pm \sqrt{\left(R2^2 - BG^2 \sin^2 v\right)}. \tag{17}$$

So the distance before handover is $CE = CG - EG$ and $\text{TBVH} = CE/\text{vel}$. We can also calculate the Network Dwell Time. To do this we need to calculate GH we know that $BH = R2$ and the angle BGH is $(180 - v)$. So again using (13)

$$GH = BG \cos(180 - v) \pm \sqrt{\left(R2^2 - BG^2 \sin^2(180 - v)\right)}. \tag{18}$$

So the dwell distance of the mobile node in the WLAN if the mobile node continues in the same direction is $EH = EG + GH$. Hence, the estimated Network Dwell Time (NDT) $= EH/\text{vel}$.

So we have examined the case of complete coverage.

4.2. Intersecting Networks. We next examine when the two networks intersect, this is shown in Figure 6.

To solve the intersecting networks scenario, we use the methods that we developed for upward and downward handovers. So we calculate CF from (14), EG and CE from (17), and GH using (18). Using this information we can calculate the distance for which the mobile node will be moving in both networks. This is denoted by $EF = CF - CE$, where $CE = CG - EG$. This gives the mobile node the opportunity to do an alternative handover at E or an imperative handover at F.

It also necessary to calculate the maximum joint coverage distance. This is given by PQ and lies along the line between

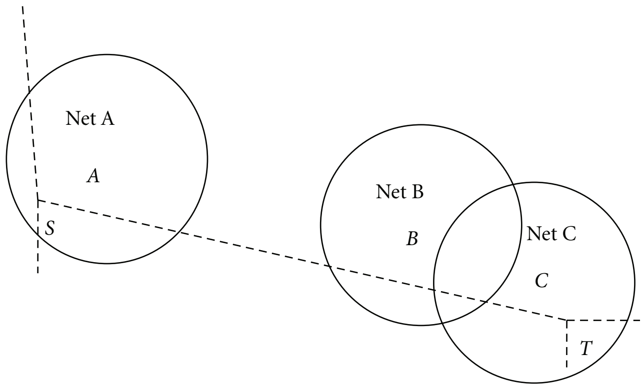


FIGURE 7: Applying the mathematical framework.

the two centres, AB. So PQ is calculated as follows: $AQ = R1$ and $AP = AB - R2$

$$PQ = AQ - AP,$$
$$PQ = R1 - (AB - R2), \quad (19)$$
$$PQ = R1 + R2 - AB.$$

4.2.1. Nonintersecting Networks. For nonintersecting networks, we use (19). So the networks are nonintersecting when the distance between the two networks is greater than the sum of the radii of the two networks as expressed in (20)

$$AB > R1 + R2. \quad (20)$$

So given that we know the position of the Access Points and the handover radius for each network as well as the position, direction, and velocity of the mobile node, it is possible to estimate the TBVH for each network as well as the estimated dwell times in each network.

5. Application of the Mathematical Framework

In the section, we show how these formulas may be applied using a given scenario below. Let us suppose that the mobile node is going along a defined path. This is shown in Figure 7 and the entire region is also covered by a single cell of a cellular network such as GSM or UMTS.

There are two junctions S and T as well as three WLAN networks as shown. Network A provides access at Junction S while Networks B and C are two other WLAN networks located along the ST route. Since each network is a WLAN network, we will take the handover radius of 100 metres which is the accepted outdoor coverage of WLAN networks [18]. The coordinates of the key points are given in Table 1. $C1$ is the point on the path where the mobile node begins to calculate TBVH and its network dwell time in Network A.

Using these coordinates we can draw a detailed diagram of the upcoming calculations based on the mathematical framework detailed above. This is shown in Figure 8.

5.1. Calculations for This Scenario

5.1.1. Calculation for Network A. Important coordinates are: $Z1 = 80, 320$, $Y1 = 280, 320$.

TABLE 1: Coordinates at key points in metres.

Symbol	X-cord	Y-cord
$C1$	40	480
S	160	280
A	180	320
B	800	200
C	960	160
T	1000	120

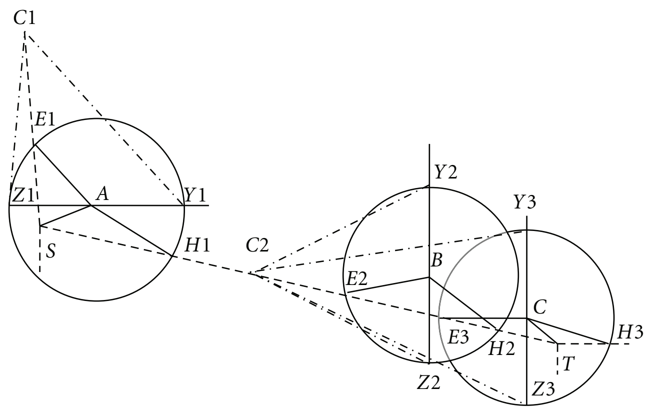


FIGURE 8: Analysis diagram.

The angle of $C1$ to the vertical, v, is given by the gradient of $C1S$. However, we can get $G1$ by simply using $C1S$ itself. So $G1 = x, 320$ where

$$\frac{(x - 40)}{(480 - 320)} = \frac{(160 - 40)}{(480 - 280)}, \quad (21)$$

so $G1 = (136, 320)$, hence $AG1 = 180 - 136 = 44$ and $v = \arctan(120/200)$

$$E1G1 = AG1\cos(90 + v) \pm \sqrt{\left(R^2 - AG1^2\sin^2(90 + v)\right)}. \quad (22)$$

Note we must use the angle $(90 + v)$ as we are using the horizontal axis, not the vertical one.

$E1G1 = 69.97143$ and $C1G1 = 186.59046$ and $C1S = 233.23808$. So TBVH distance, $C1E1 = C1G1 - E1G1 = 116.61903$ m.

We find $SH1$; we need to find the angle $ASH1$. To find this we find the angle to the horizontal of AS and the angle to the horizontal of ST to capture the change of direction

$$ASH1 = x = \arctan\left(\frac{(320 - 280)}{(180 - 160)}\right) + \arctan\left(\frac{(280 - 120)}{(1000 - 160)}\right), \quad (23)$$

Table 2: Results for Network A.

Parameter	Symbol	Value (m)	Handover
TBVH distance	$C1E1$	116.61903	Downward
Network Dwell distance	$E1H1$	219.04717	Upward

$x = 74.21925$ and $AS = 44.72136$,

$$SH1 = AS\cos(x) \pm \sqrt{\left(R^2 - AS^2\sin^2(x)\right)}, \tag{24}$$

$SH1 = 102.42812, E1S = C1S - C1E1 = 233.23808 - 116.61903 = 116.61905, E1H1 = 102.42812 + 116.61905 = 219.04717$.

The results for Network A are summarised in Table 2.

5.1.2. Calculation for Network B. We first consider $C2$ which is an arbitrary point along ST. It is at $C2$; the mobile node does calculations to determine the TBVH distance from $C2$, the expected time that will be spent in both networks and the distance that the mobile node will be under the coverage of the two networks. We have chosen the x-coordinate of $C2$ to be 480. Therefore the y-coordinate is calculated by using the gradient of ST. If $C2 = (480, y)$ then

$$\frac{(480-160)}{(280-y)} = \frac{(1000-160)}{(280-120)}. \tag{25}$$

Hence $C2 = (480, 219.04762)$.

v, which is the angle of ST to the vertical $= \arctan((1000-160)/(280-120))$.

In addition, since $G2$ is along ST if $G2 = (800, y)$, then

$$\frac{(800-160)}{(280-y)} = \frac{(1000-160)}{(280-120)}, \tag{26}$$

$G2 = (800, 158.09524)$.

Hence, $BG2 = 200 - 158.09524 = 41.90476$,

$$E2G2 = BG2\cos v \pm \sqrt{\left(R^2 - BG2^2\sin^2 v\right)}, \tag{27}$$

$E2G2 = 98.97524$.

We also calculate $G2H2$

$$G2H2 = BG2\cos(180 - v) \pm \sqrt{\left(R^2 - BG2^2\sin^2(180 - v)\right)}, \tag{28}$$

$G2H2 = 83.29346$, $C2G2 = 325.75319$.

Hence, $C2E2 = C2G2 - E2G2 = 226.77795, E2H2 = E2G2 + G2H2 = 182.2687$.

5.1.3. Calculations for Network C. We use the same technique as used for Network B to calculate $E3G3$.

So $G3 = (960, 127.61905)$.

Hence, $CG3 = 160 - 127.61905 = 32.38095$,

$$E3G3 = CG3\cos v \pm \sqrt{\left(R^2 - CG3^2\sin^2 v\right)}, \tag{29}$$

$E3G3 = 100.8649$.

Table 3: Results for Network B.

Parameter	Symbol	Value (m)	Handover
TBVH distance	$C2E2$	226.77795	Downward
Min Network Dwell distance	$E2E3$	160.98697	Horizontal
Max Network Dwell distance	$E2H2$	182.2687	Horizontal

Table 4: Results for Network C.

Parameter	Symbol	Value (m)	Handover
Min Network Dwell distance	$H2H3$	171.95314	Upward
Max Network Dwell distance	$E3H3$	193.23487	Upward

We first calculate $G3T$ since we know the coordinates of $G3$ and T:

$$G3T = 40.71916. \tag{30}$$

So we can find CT

$$CT^2 = G3T^2 + CG3^2 - 2G3T * CG3\cos(180 - v), \tag{31}$$

$CT = 56.56969$.

We find the angle $CTH = (180 - \arccos(40/CT))$.

So we can now find $TH3$

$$TH3 = CT\cos CTH \pm \sqrt{\left(R^2 - CT^2\sin^2 CTH\right)}, \tag{32}$$

$TH3 = 51.65081$ m.

We also would like to find the overlap distance between the two networks.

We can do so by finding $G2G3$ which we can find directly because we know the coordinates of $G2$ and $G3$:

$$G2G3 = 162.87663. \tag{33}$$

So $G2E3 = G2G3 - E3G3 = 162.87663 - 100.8649 = 62.01173$.

So the intersection distance $= G2H2 - G2E3 = 83.29346 - 62.01173 = 21.28173$.

Finally the Maximum Dwell distance in Network C $= E3H3 = E3G3 + G3T + TH3 = 193.23487$.

The Minimum Dwell distance occurs if the mobile node delays handover until it has left Network B.

So Minimum Dwell distance in Network C = 193.23487 − 21.28173 = 171.95314.

Similarly the Minimum Dwell distance in Network B is 182.2687 − 21.28173 = 160.98697.

So the results of networks B and C are shown in Tables 3 and 4.

The results for Network A, B, and C are summarised in Table 5.

5.2. Checking the Results. In order to check these results, it was decided to draw the whole system to scale using a large sheet of graph paper and hence the results above were compared with the values on the graph. A scale of 1 mm to 4 m was used. The results compared favourably with the graphical results. A computer program [19] was also used to

Table 5: Results for Networks A, B, and C for HR = 100 m.

Network	Parameter	Symbol	Value (m)	Handover
A	TBVH dist	$C1E1$	116.61903	Downward
	ND dist	$E1H1$	219.04717	Upward
$B \cap C$	Intersect dist	$E3H2$	21.28173	Alternative
B	TBVH dist	$C2E2$	226.77795	Downward
	Min ND dist	$E2E3$	160.98697	Horizontal
	Max ND dist	$E2H2$	182.26870	Horizontal
C	Min ND dist	$H2H3$	171.95314	Upward
	Max ND dist	$E3H3$	193.23487	Upward

Table 6: Results for Networks A, B, and C for HR = 104 m.

Network	Parameter	Symbol	Value (m)	Change (m)
A	TBVH dist	$C1E1$	112.3	−4.38
	ND dist	$E1H1$	227.7	7.95
$B \cap C$	Intersect dist	$E3H2$	29.90	8.62
B	TBVH dist	$C2E2$	222.4	−4.38
	Min ND dist	$E2E3$	161.1	0.11
	Max ND dist	$E2H2$	191.1	8.83
C	Min ND dist	$H2H3$	171.9	−0.05
	Max ND dist	$E3H3$	201.8	8.57

check these values. The program was done in C# and read the location of the mobile node, the location of the different networks as well as the handover radius from a table in an SQL database and calculated the results for different points of the path as the mobile moved along the route. This verifies that this approach is viable. So the mobile node can calculate when and where to handover given that the mobile knows where the centre of each network is located and the handover radius of each network as well as the direction and speed of the mobile node [20].

6. Results for Different Handover Radii

6.1. Using Propagation Models to Get a More Accurate Handover Radius. As previously described, in order to get a more accurate estimation of the handover radius, we need to have the right propagation model as well as know the threshold value at which to handover. This is in fact dependent on the type of equipment being used and the power at the transmitter. To illustrate the issues, we imagine a scenario where the transmission power is 100 milliwatts, and the range of detection of the the WLAN receiver is chosen as −82 dB. Using the standard Free Space Model, the MATLAB tool was used to calculate the handover radius for this scenario. The result was a handover radius of 104 metres [21].

So we again calculate the values for the situation shown in Figure 8. The results are shown in Table 6. These results show that the TBVH times decrease while the Network Dwell distance and the intersect distance increase, with the latter increasing by over 40 percent. Thus small changes in the handover radius can affect handover decisions. This is also more important for 3G and UMTS systems where different propagation models will produce different results. For these systems, it is assumed that the mobile node will use a threshold of around −120 dB [22]. The handover radius for urban and suburban Hata models are given in Table 7. These results again highlight the fact that if a wrong model is used then the values of TBVH and NDT would not be very accurate and therefore the proposed solution would not be useful.

These results show that in order to really build an efficient proactive handover system which can be managed in an autonomous manner, it is necessary to have more accurate information about the networking infrastructure in the local area including the location of the Access Points/Base-stations as well as the power being transmitted at these locations plus a propagation model that represents the wireless network characteristics in that local area. Using this information, it would be possible to calculate the handover radius. This information along with the location and velocity of mobile node plus the dB thresholds for the wireless interfaces on the mobile device will make it possible to do effective vertical handover using the mathematical framework shown above.

Table 7: Handover radius for 3G, UMTS at −120 dB.

Propagation model	Handover radius (m)
Urban Hata Model	1850
Suburban Hata Model	1980

7. Information on Networks in the Local Area

The approach presented shows that proactive handover can be performed over several networks that may be a few kilometres from the mobile node. In this context, it is not possible for the mobile node to deploy its network discovery mechanisms via its wireless network interfaces to find all the related networks. Hence information on networks in the local area needs to be stored in the core network.

7.1. Using IEEE 802.21. Such a facility has been defined in the IEEE 802.21 standards which specify Media Independent Handover (MIH) mechanisms that can be used to manage several wireless networks [23]. The MIH specifies 3 main mechanisms.

(i) Media-Independent Event Service or MIES: this service is used to relay events from low-level device interfaces to the upper layers via the MIH service layer. Applications register with the MIH layer the events in which they are interested. The MIH layer is, in turn, sent notification of events occurring at the lower level interfaces. Events include Link_Detected, Link_Up, Link_Down, Link_Going_Down, Link_Hand-over_Imminent, and Link_Handover_Complete. The MIH layer then sends MIH event notifications to applications which want to be notified about specific events.

(ii) Media-Independent Command Service or MICS: this allows applications to send commands to the low-level interfaces via the MIH layer. Applications can use MICS commands to find out the status of various interfaces. Commands sent to MIH layer by the upper layers include MIH_Link_Get_Parameters which gets the link status; the MIH_Link_Configure_Thresholds command is used to configure thresholds on mobile devices. Commands are also used to initiate handover to different target networks. Where appropriate, MIH commands are then converted to Link Commands which are sent to the relative network interfaces.

(iii) Media-Independent Information Service or MIIS: this service is used to allow mobile nodes to obtain network information within a geographical area so as to facilitate network selection and smooth handover.

7.1.1. Network Information Provided by the MIIS Service. Network information provided by the MIIS is divided into a number of containers. A higher level container can have lower level containers embedded within it. These are specified below.

(i) IE_CONTAINER_LIST_OF_CONTAINERS: at the highest level, there is a container for the list of heterogeneous networks in a given area.

(ii) IE_CONTAINER_NETWORK: each network is specified using a network container which gives all the information that depicts the access network including the number of POAs, IP subnet address ranges, and so forth.

(iii) IE_CONTAINER_POA: this contains all the information for a given POA.

The IE_CONTAINER_POA contains specific information on a given POA including its Media Access Control (MAC) address, Range of Channel, IP address, Geographical Location, System, and information on subnets. Some POA containers may also contain vendor-specific elements.

7.2. A Prototype Database for Intelligent Proactive Handover. It was decided to develop a prototype database on the networks in the local area [24] to allow mobile nodes to query a database server to get the required information that would allow them to calculate where and when is the best time to do the handover. The database was used to specify the WLAN networks in the local area. It also provided information on the parameters of the Free Space Model that had to be used in the local area, using the equation below

$$\mathrm{RSS}_p = E_t - 10\beta\log_{10} l_{op} + \xi, \tag{34}$$

where RSS_p is the received signal strength measured in mW, E_t is the transmission power of the AP in mW, β is the path loss component whose value is between 2 and 4, l_{op} is the distance of the mobile node from the Access Point, and ξ is the Gaussian distributed random variable.

Thus if this model is used, the values of β and ξ must be calculated for every local context and thus these values need to be stored in a local database. The stored parameters are shown in Table 8.

Table 8: Network information database.

Parameter in database	Data type of the field
Latitude	double
Longitude	double
Channel number	int
Coverage radius	int
Transmit power	int
Mac address	Varchar (50)
Path loss exponent	int
Gaussian distribution exponent	int

The program was written in Java and showed that the mobile node could read the information from the database and then was able to calculate the handover radius, given the values of the β and ϵ. Based on these values it can then calculate TBVH and estimate NDT. It is able recalculate these values if the mobile node changes direction.

7.3. Determining the Exit Radius. In Section 3 we defined the exit radius as

$$R_E \leq R_H - (\mathrm{vel} * T_{EH}). \tag{35}$$

Thus the exit radius, R_E, is very dependent on the handover radius, the velocity of the Mobile node, vel, and the time it takes to effect the handover, T_{EH}. The time taken to effect a handover was shown to be dependent by various factors as discussed in [25].

(i) *The Detection Time, t_d*. This is the time taken to detect the first signal of the new network. This can be done by detecting Router Advertisements or by the use of L2 triggers.

(ii) *The Configuration Time, t_c*. This is the time from detecting the network to the time taken by the mobile node to get and configure its Care-of-Address (COA).

(iii) The Registration Time, t_r. This is the time between the delivery of the Binding Update to the Home Agent and corresponding nodes and the time a packet arrives on the new interface.

(iv) *The Adaptation Time, t_a*. For vertical handovers, this is the time taken by the mobile node to adapt the connection to the new technology at the transport layer by adjusting the TCP state machine parameters (e.g., congestion window size, timeout timers, etc.), due to differences in the link characteristics.

For reactive handover, all four times must be added together since reactive handovers respond to network conditions as discussed in [11]. The mobile device therefore knows nothing beforehand about the characteristics of the various networks. This is not the case with an MIH-enabled device which can find out about networks via the Media-Independent Information Service. For an MIH-enabled device, there is no detection time since the mobile node

Table 9: Delay for proactive handover scenarios.

Source	Target	Handover	t_r (s)	t_a (s)	T_{EH} (s)
802.11g	HSDPA	Upward	3	2	3
HSDPA	802.11g	Downward	4	0	4
802.11g	802.11g	Horizontal	4	0	4
HSDPA	HSDPA	Horizontal	1	0	1

Table 10: Exit radius for different speeds in metres.

Network	Type	T_{EH} (s)	R_E 5 Kmph	R_E 50 Kmph
A, C	Upward	3	95.83	58.33
B	Horizontal	4	94.4	44.44

would know where all the local networks are located. There is also a negligible delay for the configuration time since the mobile node will know the IP address of the target network. The Registration Time is still valid. In addition, for proactive networks the need for the transport protocol to adapt can be signalled before or during handover and not after the handovers occur. So it means that the adaptation time can be done in parallel with the registration time. So for proactive handover,

$$T_{EH} = \mathrm{MAX}(t_r, t_a). \tag{36}$$

Using the measured results rounded up to the nearest second [11], in Table 9, we summarize the results for handover delay for proactive handovers between WLAN and 3G in our scenario.

Table 10 shows the exit radius for the different WLAN networks for a handover radius of 100 meters. We are interested in typical walking and driving speeds, so values of 5 Kilometres per hour (Kmph) and 50 Kmph were selected. These results, which are given in metres, show that the exit radius is very dependent on velocity of the mobile node.

7.4. Using Community-Based Local Information Systems. In current networking environments, acquiring information about the local environment may be difficult for many reasons. As pointed out previously, cellular networks are carefully planned and detailed configuration characteristics may be regarded by some mobile operators as commercially sensitive information. WLAN systems may be set up on a per-home or per-person basis. This means that it is very unlikely that these networks would be registered with some central authority unless it is required by law.

Another way of gathering information about coverage and signal strengths is to allow users to access information on coverage data which was gathered from previous journeys along the same route or by other mobile nodes [26]. This has been explored in the concept of mobile virtual communities, which share aggregated data about mobile movement around particular nodes. The mechanism by which this data is stored is known as *Wireless Footprinting.*

Figure 9 shows a user with a Smart Phone walking through a number of wireless networks via a defined route/path. There is a Wireless Footprint Server (WFS) which

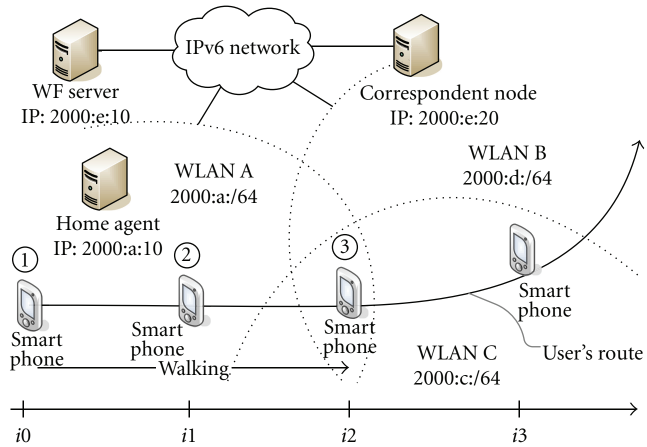


Figure 9: An example of Wireless Footprinting.

stores information about the wireless networks that the user has previously encountered at the present location. The Smart Phone contacts the WFS to find out if there is any wireless data associated with its current location. If so, it downloads the data and puts it in a local cache on the client node. The client can also ask for an aggregated summary of wireless characteristics in the local area based on the context information given to the server by other mobile nodes. The server therefore provides some key functions including:

(i) Storing client details: this mechanism allows the client to store details about wireless characteristics for various networks including signal strength, Access Point/Base-station IDs, bandwidth, QoS, and so forth, at a given location.

(ii) Context retrieval and storage: this allows the mobile node to retrieve context information about all the networks in the local area. This is similar to the MIIS service in IEEE 802.21.

(iii) Feedback search: this allows the server to provide information on wireless characteristics of local networks.

(iv) Cache consistency: the server pushes updates to the client so as to keep the local cache up-to-date.

(v) Handover history: the server can also record when and where the mobile node handed over. Though at times this may be useful, efficient handover depends on many real-time factors to characterize the state of the network.

It should be pointed out that a small experimental system [27] was built and the result showed improved connectivity during handover periods which indicates that better handover management was achieved.

7.5. Towards a Dynamic Propagation Model. The authors believe that the idea of community-based, context-aware information can also be used to develop more accurate and dynamic propagation models to characterize wireless characteristics in the local area. This is because the Wireless

Footprinting Server or (WFS) can use the data on signal strengths at each location to build an accurate model. For WLAN this would mean finding the local parameters for the Free Space Model. For cellular networks the Hata models would be more accurately tuned. Thus using this approach an efficient proactive handover mechanism can be developed. Since the WFS is being updated frequently, it would be possible to calculate the most relevant propagation model in an adaptable and dynamic way which could take into account weather, and so forth. Ideally, therefore, it is necessary to have static information about the networking infrastructure as well as dynamic information which will be changing in real-time and can be used to make effective handover decisions.

7.6. Towards a Comprehensive Handover Framework. As shown above, contextual information can be used for effective imperative handover. However, the use of contextual information has also been used to look at alternative handover [28]. This means that the use of both location and contextual information along with the mechanisms presented above can be used to build a more comprehensive framework to manage vertical handover.

8. The Need for a New Architecture for Mobile Networks

This work above has shown that in order to build an efficient system, we will have to be able to integrate several new mechanisms and hence a framework is needed to do this. The main motivation for this is the belief that future mobile systems which will support heterogeneous networking must bring together mechanisms that support communication, mobility, QoS, and security in an integrated fashion. The Y-Comm architecture is an architecture designed to address this goal. Y-Comm is based around a vision of the future predicated on two key assumptions: the first is that mobile devices will have several wireless interfaces and so must function in an increasing heterogeneous environment. The second is that the development and deployment of these wireless networks will point to a significant change in the architecture of the Internet. The Internet will no longer be represented by a single monolithic entity but by a superfast core composed of an optical backbone and fast access net-works while the end networks will be dominated by the deployment of wireless technologies. This is shown in Figure 10. Other network architectures for future mobile systems have been explored such as Ambient Networks [29] and Mobile Ethernet [30] but Y-Comm has been found to have a more open approach compared to these systems [31].

Y-Comm [32] therefore proposes two frameworks to represent this new reality. The first is called the *Peripheral Framework* and deals with operations and functions on the mobile node. The other framework is called the *Core Framework* and shows the functionality required in the core network to support the Peripheral Framework. The structure of the Y-Comm architecture is shown in Figure 11. A brief explanation of Y-Comm is now attempted starting with the lowest layer. A more detailed explanation can be found in [7, 33].

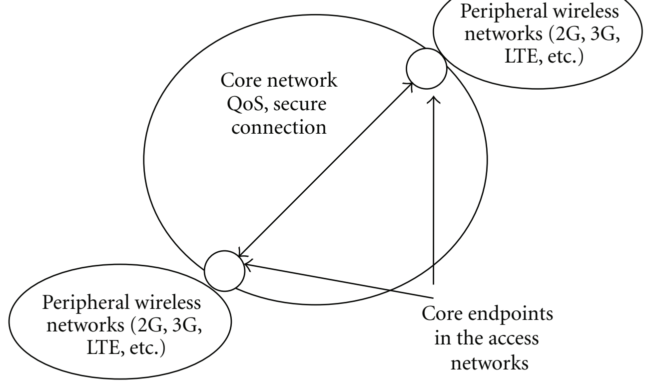


Figure 10: The future Internet structure.

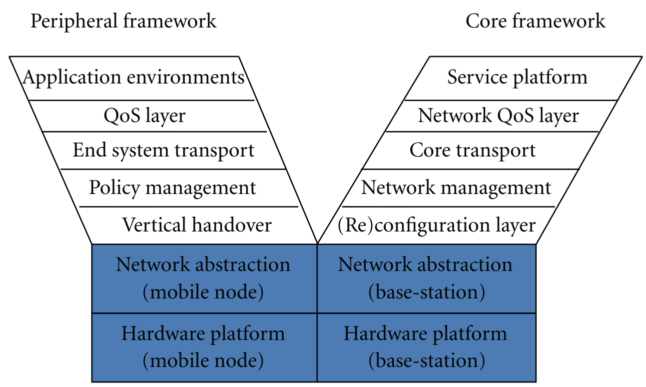


Figure 11: The Y-Comm framework.

8.1. The Peripheral Framework. The *Hardware Platform Layer* (HPL) is used to classify all relevant wireless technologies. Hence different wireless technologies which are characterised by the electromagnetic spectrum, MAC, and modulation techniques make up this layer. The *Network Abstraction Layer* (NAL) provides a common interface to manage and control all the wireless networks. These first two layers for both frameworks are similar in functionality. In the Peripheral Framework, the Hardware Platform and the Network Abstraction layers run on the mobile to support various wireless network technologies while in the Core Framework these two layers are used to control the functions of base stations of different wireless technologies.

The *Vertical Handover Layer* (VHL) executes vertical handover. So this layer acquires the resources for handover, does the signalling, and context transfer for vertical handover. The *Policy Management Layer* (PML) decides whether and when handover should occur. This is done by looking at various parameters related to handover such as signal strength and using policy rules to decide both the time and place for doing the handover.

The *End Transport Layer* (ETL) is used to provide network and transport functions to the mobile nodes in

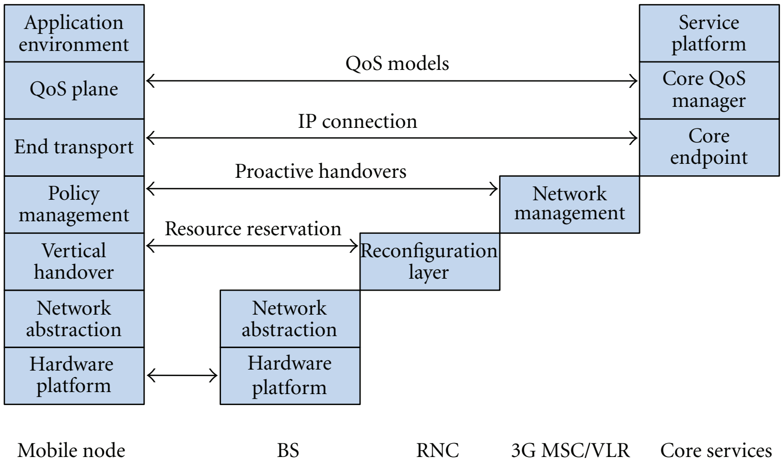


FIGURE 12: Mapping of Y-Comm unto 3G/UMTS.

peripheral networks. It allows the mobile node to make end-to-end connections across the core network. The *QoS Layer* (QL) in the Peripheral Framework supports two mechanisms for handling QoS. The first is defined as Downward QoS. This is where an application specifies its required quality-of-service to the system and the system attempts to maintain this QoS over varying network channels. The other definition is Upward QoS where the application itself tries to adapt to the changing QoS. This layer also monitors the QoS used by the wireless network as a whole to ensure stable operation. The final layer of the Peripheral Framework is called the *Applications Environments Layer* (AEL). This layer specifies a set of objects, functions, and routines to build applications which make use of the framework.

8.2. The Core Framework. As previously mentioned, the first two layers of the Core Framework are engaged in controlling base-station operations. The third layer is called the *Reconfiguration Layer* (REL). It is a control plane to manage key infrastructure such as routers, switches, and other mobile network infrastructure using programmable networking techniques [34]. *The Network Management Layer* (NML) is a management plane that is used to control networking operations in the core. This layer can divide the core into a number of networks which are managed in an integrated fashion. It also gathers information on peripheral networks such that it can inform the Policy Management Layer running on mobile nodes about wireless networks at their various locations. In the context of the work already presented in this paper, the Wireless Footprinting Server (WFS) can be viewed as a manifestation of the Network Management Layer.

The next layer, called the *Core Transport System* (CTS), is concerned with moving data through the core network. Where the peripheral networks join the core network is called a *core endpoint*. Core endpoints are usually situated in access networks and several peripheral networks may be attached to a single-core endpoint. CTS is concerned primarily with moving data between core endpoints with a given QoS and a specified level of security.

The *Network QoS Layer* (NQL) is concerned with QoS issues within the core network especially at the interface between the core network and the peripheral networks. Finally the *Service Platform Layer* (SPL) allows services to be installed on various networks at the same time.

Though Y-Comm is a new architecture; it can be mapped onto current mobile infrastructure such as 3G or GSM as shown in Figure 12. As stated previously, the Peripheral Framework runs on the mobile node, while the Core Framework is distributed throughout the 3G/UMTS network infrastructure as shown.

Issues of security and quality-of-service have been investigated in the context of the Y-Comm architecture and new concepts such as Targeted Security Models have been developed [35, 36].

8.3. Proactive Handovers in Y-Comm. This section looks at proactive handovers in Y-Comm. This is shown in Figure 13. In order to perform vertical handover using a mathematical model approach, it is necessary to know the topology of these local networks. In Y-Comm, this information is managed by the Network Management Layer in the Core Framework. The mobile node therefore polls the NML to obtain information with regard to all local wireless networks, their topologies, and QoS characteristics. This information along with the direction and speed of the mobile as well as the QoS of ongoing connections is used by the Policy Management Layer to determine where and when handover should occur. The PML calculates TBVH—the period after which handover will occur as well as the estimated Network Dwell Time. This information is communicated to the Vertical Handover Layer which immediately requests resources to do a handover. Even though the resources are acquired early, handover actually takes place when TBVH expires.

In addition, once the PML decides to handover, the new IP address, the new QoS as well as TBVH, and estimated Network Dwell Time are communicated to the upper layers. Given TBVH, the upper layers are expected to take the necessary steps to avoid any packet loss, latency, or slow

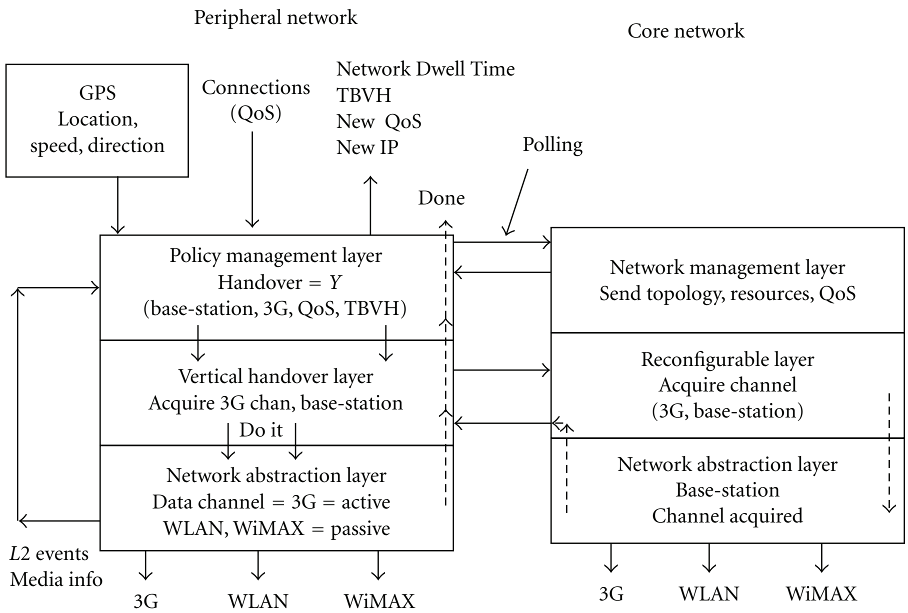


Figure 13: Proactive handover in the Y-Comm architecture.

adaptation. For example, it may be possible for the End-Transport Layer to signal an impending change in the QoS on current transport connections and to begin to buffer packets ahead of the handover. After handover, the previous channel used by the mobile node is released.

8.4. Towards an Ontology for Proactive Handover in Y-Comm. Ontologies are not only an elegant solution for modelling a problem. Ontologies are developed to formally describe semantic meanings of terms and their relationships in a domain. Heterogeneous entities working in a domain have semantic interoperability when they exchange the right information. Practices such as modularity, extensibility, and formal definitions of axioms can positively impact on the understanding and utilisation of systems mapped by ontologies. To accept the definitions of an ontology, explicitly or implicitly, means to be in agreement with the semantic representation of concepts and their relationships.

In order to develop a real system, it is necessary to define the functions and mechanisms in a way that will allow us to integrate different mechanisms and technologies. An ontology is a definition of a system with a detailed specification of each component. Ontologies will be used to describe the Y-Comm framework with a detailed specification of each component. The ontologies will assist in semantic interoperability between providers and users using Y-Comm. The definitions of the ontology will be stored in the ontology itself written in OWL (Web Ontology Language). The actual data will be stored in a database using the Resource Description Framework or RDF. However, the terms of the ontology are the descriptions of the data in the database [37].

An ontology for proactive in Y-Comm has been developed and is shown in Figure 14. It describes the relationships between different parts of the proactive handover system. Using this ontology, prototype code can be generated in Java showing how the functions for different parts of the lower layers in Y-Comm could be implemented. This is shown in Figure 15. Hence this approach represents a possible implementation path for future mobile networks.

9. Previous Work

Handover issues have been investigated for some time and it would be difficult to give a full summary of all the efforts. Original work was centred around horizontal handover in cellular networks and was done by mobile operators. With the introduction of Mobile IPv4 and Mobile IPv6 [5], client-based handover began to be investigated. For these mechanisms, handover latency is high because they only work at the network level as they are based on Router Advertisements (RAs) which are relatively slow. Fast Mobile IPv6 (FMIPv6) [6] was developed to make use of L2 events and triggers to reduce handover latency.

Reactive handover was studied in a number of simulations and in the Cambridge Wireless Testbed [38]. The testbed used the Vodafone 3G experimental network, with home and foreign WLANs and a wired IPv6 LAN. Using the testbed, PROTON [39], a policy manager for reactive handovers was developed. PROTON was implemented using a 3-layer structure.

However, with the rise of heterogeneous networking there was more interest in handover prediction techniques. This was explored using three research methods.

(i) History-Based Prediction.

(ii) Coverage-Based Prediction.

(iii) Mathematical-Modelling-Based Prediction.

9.1. History-Based Prediction. This prediction is based on the past movements of the mobile device in the network.

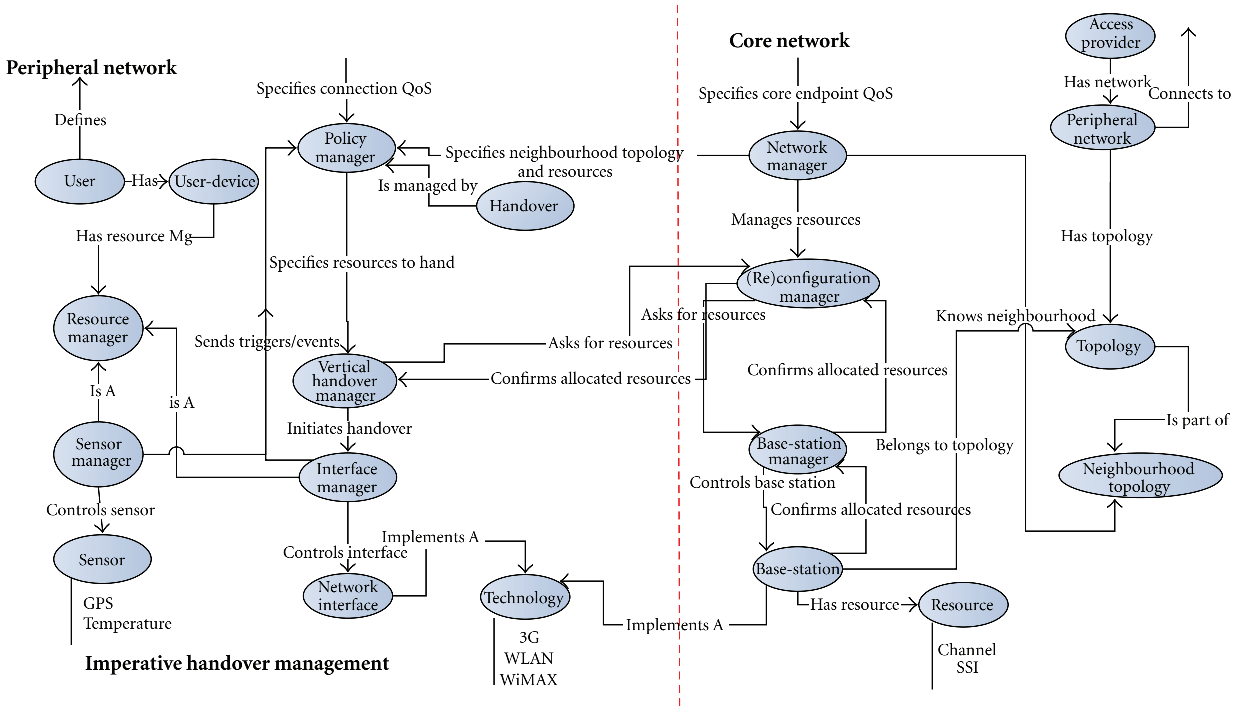


Figure 14: An ontology for proactive imperative handover in Y-Comm.

```java
//Get the geographical coordinates
String latitude = request.getParameter("latitude");
String longitude = request.getParameter("longitude";
// Y-COMM ontology location
String ontologyFile = "file:y-comm_ontology.owl";
String ontologyInstancesFile = "file:y-comm_ontology_instances.owl";
//Create the models for the ontology and for the instances of the
// ontology from the database
OntModel ontologySchema = ModelFactory.createOntologyModel();
ontologySchema.read(ontologyFile);
ontModel ontologyData = ModelFactory.createOntologyModel();
ontologyData.read(ontologyInstancesFile);
//Create a list with the rules
String ruleFile = "file.feedbackService.rules";
List rule = com.hp.hpl.jena.reasoner.rulesys.Rule.rulesFromURL(ruleFile);
//Create a reasoner with the rules
Reasoner reasoner = new GenericRuleReasoner(rule);
reasoner = reasoner.bindSchema (ontologySchema);
//Create a inference model
InfModel infModel = ModelFactory.createInfModel (reasoner, ontoloyData);
//Retrieve and print all context data at the geographical point
PrintWriter out = response.getWriter();
printTopology(ontologyData, latitude, longitude, ontologyData, out);
```

Figure 15: Code generated from the Y-Comm ontology.

Previously stored data is helpful in mobile node movement prediction. The main disadvantage with this approach is it has to depend on a lot of data before making predictions.

9.2. Coverage-Based Prediction. With this technique, handover prediction is based on the knowledge of network coverage which is maintained in a database. So the database contains coverage information instead of records of the movement of the mobile node. Coverage data must be frequently updated for this scheme to work accurately and it is a difficult process maintaining large databases. Soh and Kim [40] proposed a proactive mobility prediction technique. In this technique, horizontal handover time is predicted using the road topology and the location of the mobile node. Recent studies [41] proved that for better network performance calculating the time before handover is worthwhile. The same drawback exists here also in that it is not possible to predict the path and time of mobile node if it takes alternate path instead of the stored one. Cottingham et al. [42] used coverage maps for mapping network coverage. Received signal strength is measured by driving on roads and thus the real-time data is gathered. But the problem is that to gather all the information takes a lot of time which is undesirable.

9.3. Mathematical-Modelling-Based Prediction. In this technique, models and mathematical formulas are used to predict handover conditions. The drawback of this technique is that it can be computationally expensive and the resources of mobile node may be limited. Saleh [43] proposed an algorithm for making location-aided handover decisions. Unwanted vertical handovers are predicted by this algorithm. Another model for predicting unwanted handovers is proposed in [44] and uses a mathematically derived model. Circular WLAN coverage is assumed by this model for calculating the distance between the mobile node and the Access Point, along with received signal strength. This approach fails if the mobile node enters the WLAN network and changes its direction.

This paper shows how the limitations previously mentioned can be overcome. It has shown how to calculate TBVH and how to estimate NDT. It shows how these values can be calculated even when there is a change of direction of the mobile node. This study of proactive handovers has also indicated the need to implement location-based and context-aware services that keep mobile nodes informed of the networks in the local area.

10. Conclusions and Future Work

This paper has explored a set of mechanisms which can be used to build a comprehensive framework to support proactive handover in heterogeneous networks. A mathematical framework was first developed and mechanisms to ensure its accuracy using location and contextual information were investigated. Finally, it has been demonstrated how these mechanisms can be integrated into the Y-Comm architecture and how using an ontology for proactive handover provides an implementation path for future systems.

The work presented in this paper was based around the concept of a handover radius. Though this is relevant in most outdoor environments, more sophisticated techniques of defining coverage and the mathematics required to model irregular coverage spaces are being explored. Using the ontological techniques explained above, work has begun to use IEEE 802.21 mechanisms to implement an imperative handover management based on the lower layers of the Y-Comm architecture. This will form the basis of a comprehensive handover framework [37]. Work is also being done to bring up an experimental testbed based on IEEE 802.21. In addition, an investigation is beginning into how to use estimated TBVH and Network Dwell Times to improve channel allocation strategies.

References

[1] J. Schiller, *Mobile Communications*, Addison Wesley, 2nd edition, 2003.

[2] J. McNair and F. Zhu, "Vertical handoffs in fourth-generation multinetwork environments," *IEEE Wireless Communications*, vol. 11, no. 3, pp. 8–15, 2004.

[3] E. Gustafsson and A. Jonsson, "Always best connected," *IEEE Wireless Communications*, vol. 10, no. 1, pp. 49–55, 2003.

[4] IEEE 802.21 Working Group, "IEEE 802.21: Media Independent Handover Services," http://www.ieee802.org/21.

[5] D. Johnson, C. Perkins, and J. Arkko, *RFC 3775 —Mobility Support in IPv6*, IETF, 2004.

[6] P. McCann, *Mobile IPV6 Fast Handovers for 802.11 Networks*, IETF, 2005.

[7] G. Mapp, D. Cottingham, F. Shaikh et al., "An architectural framework for heterogeneous networking," in *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS '06)*, pp. 5–10, August 2006.

[8] S. Bansal, A. Lasebae, and R. Comley, "Freedom to choose along-with freedom of mobility," in *Proceedings of the International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA '04)*, p. 191, April 2004.

[9] L. Patanapongpibul, G. Mapp, and A. Hopper, "An end-system approach to mobility management for 4G networks and its application to thin-client computing," *ACM Mobile Computing and Communications Review*, vol. 10, no. 3, pp. 13–33, 2006.

[10] D. Cottingham, I. Wassell, and R. Harle, "Performance of IEEE 802.11a in vehicular contexts," in *Proceedings of IEEE 65th Vehicular Technology Conference (VTC '07)*, pp. 854–858, April 2007.

[11] D. Cottingham, *Vehicular wireless communications [Ph.D. thesis]*, Computer Laboratory, University of Cambridge, January 2009.

[12] F. Shaikh, A. Lasebae, and G. Mapp, "Proactive policy management for heterogeneous networks," in *Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA '08)*, April 2008.

[13] F. Shaikh, *Intelligent proactive handover and QoS management using TBVH in heterogeneous networks [Ph.D. thesis]*, School of Engineering and Information Sciences, Middlesex University, January 2010.

[14] W. Chen, J. Liu, and H. Huang, "Implementation and performance study of IEEE 802.21 in integrated IEEE 802.11/802.16e

networks," in *Proceedings of the 10th International Conference on Parallel and Distributed Systems*, 2004.

[15] V. Abhayawardhana, I. Wassell, D. Crosby, M. Sellars, and M. Brown, "Comparison of empirical propagation path loss models for fixed wireless access systems," in *Proceedings of IEEE 61st Vehicular Technology Conference (VTC '05)*, pp. 73–77, Springer, June 2005.

[16] "Radio theory and link planning for Wireless LAN (WLAN)," http://www.swisswireless.org/wlan_calc_en.html.

[17] G. Mapp, F. Shaikh, M. Aiash, R. Vanni, M. Augusto and E. Moreira, "Exploring efficient imperative handover mechanisms for heterogeneous networks," in *Proceedings of the International Symposium of Emerging Ubiquitous and Persuasive Systems*, Indianapolis, Ind, USA, August 2009.

[18] Wikipedia, "Article on WiFi," http://en.wikipedia.org/wiki/Wi-Fi.

[19] J. Medisetty, *Exploring coverage areas to facilitate vertical proactive handovers [M. Sc. thesis]*, School of Engineering and Information Sciences, Middlesex University, November 2011.

[20] S. Maniyar, *Characterising network attachment points to facilitate proactive vertical handover [M. Sc. thesis]*, School of Engineering and Information Sciences, Middlesex University, November 2011.

[21] N. Chinnam, *Exploring different propagation models for wireless networks for different environments [M. Sc. thesis]*, School of Engineering and Information Sciences, Middlesex University, February 2012.

[22] M. Braga, "Cell phone reception explained: what the bars mean," July 2010, http://www.tested.com/news/cell-phone-reception-explained-what-the-bars-mean/510/.

[23] V. Andrei, E. C. Popovici, O. Fratu, and S. V. Halunga, "Development of an IEEE 802.21 media independent information service," in *Proceedings of the 17th IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR '10)*, pp. 123–128, May 2010.

[24] D. Dave, *Local network information for proactive handovers [M. Sc. thesis]*, School of Engineering and Information Sciences, Middlesex University, August 2010.

[25] D. Cottingham and P. Vidales, "Is latency the real enemy in next generation networks?" in *Proceedings of the 1st International Workshop on Convergence of Hetergeneous Wireless Networks*, July 2005.

[26] R. R. F. Lopes, B. J. van Beijnum, and E. D. S. Moreira, "Towards a feasible social-based methodology to manage wireless connectivity context data," in *Proceedings of IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '11)*, pp. 237–244, Shanghai, China, October 2011.

[27] R. R. F. Lopes, B. J. Van Beijnum, and E. D. S. Moreira, "Managing wireless IP-connectivity experiences as mobile social media," in *Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT '10)*, pp. 154–161, October 2010.

[28] E. D. S. Moreira, J. Crowcroft, G. Mapp, D. Cottingham, P. Hui, and R. Vanni, "Exploiting contextual handover information for versatile services in NGN environments," in *Proceeings of the 2nd International Conference on Digital Information Management (ICDIM '07)*, October 2007.

[29] U. Horn, C. Prehofer, H. Karl et al., "Ambient networks: an architecture for communication networks beyond 3G," *IEEE Wireless Communications*, vol. 11, no. 2, pp. 14–22, 2004.

[30] M. Kuroda, M. Yoshida, R. Ono, S. Kiyomoto, and T. Tanaka, "Secure service and network framework for mobile ethernet," *Wireless Personal Communications*, vol. 29, no. 3-4, pp. 161–190, 2004.

[31] M. Aiash, G. Mapp, A. Lasebae et al., "A survey of potential architectures for communication in heterogeneous networks," in *Proceedings of the Wireless Telecommunications Symposium (WTS '12)*, April 2012.

[32] Y-Comm Web Page, http://www.mdx.ac.uk/research/areas/software/ycomm_research.aspx.

[33] G. Mapp, F. Shaikh, J. Crowcroft, D. Cottingham, and J. Baliosian, "Y-Comm: a global architecture for heterogeneous networking," in *Proceedings of the 3rd Annual International Wireless Internet Conference (WICON '07)*, October 2007.

[34] J. der Merwe and I. Leslie, "Switchlets and dynamic virtual ATM networks," in *Proceedings of the Integrated Network Management V*, 1997.

[35] M. Aiash, G. Mapp, and A. Lasebae, "Security and QoS integration for protecting service providers in hterogeneous environments," *IAENG International Journal of Computer Science*, vol. 38, no. 4, pp. 384–393.

[36] M. Aiash, G. Mapp, A. Lasebae, J. Loo, and R. Phan, "A formally verified AKA protocol for vertical handover in heterogeneous environments using Casper/FDR," *EURASIP Journal on Wireless Communincations and Networking*, 2012. In press, http://jwcn.eurasipjournals.com/content/2012/1/57/.

[37] R. Vanni, K. Branco, E. Moreira, and G. Mapp, "Information for handover management in heterogeneous networks: data representation, languages and integrated platforms," in *Proceedings of the 7th CONTECSI-International Conference on Information Systems and Technology Management*, pp. 1–13, Sao Paulo, Brazil, May 2010.

[38] P. Vidales, G. Mapp, F. Stajano, J. Crowcroft, and C. J. Bernardos, "A practical approach for 4G systems: deployment of overlay networks," in *Proceedings of the 1st International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom '05)*, pp. 172–181, IEEE Computer Society, February 2005.

[39] P. Vidales, R. Chakravorty, and C. Policroniades, "PROTON: a policy-based solution for future 4G devices," in *Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY '04)*, pp. 219–222, June 2004.

[40] W. Soh and H. S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions," in *Proceedings of IEEE 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2766–2777, March 2004.

[41] F. Shaikh, G. Mapp, and A. Lasebae, "Proactive policy management using TBVH mechanism in heterogeneous networks," in *Proceedings of the International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST '07)*, pp. 151–157, September 2007.

[42] D. Cottingham, R. Harle, and A. Hopper, "Constructing accurate, space-efficient, wireless coverage maps for vehicular contexts," in *Proceedings of the 4th International Wireless Internet Conference (ISTWICON '08)*, November 2008.

[43] A. Saleh, *A location-aided decision algorithm for handover across heterogeneous wireless overlay networks [M.Sc. thesis]*, Virginia Polytechnique Institute and State University, 2004.

[44] X. Yan, N. Mani, and Y. Cekercioglu, "A traveling distance prediction based method to minimize unnecessary handovers from cellular networks to WLANs," *IEEE Communications Letters*, vol. 12, no. 1, pp. 14–16, 2008.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 493261, 10 pages
doi:10.1155/2012/493261

Research Article

Enhancing Existing Communication Services with Context Awareness

Bachir Chihani,[1,2] Emmanuel Bertin,[1,2] Irsalina Salsabila Suprapto,[3] Julien Zimmermann,[1] and Noël Crespi[2]

[1] *Orange Labs, 42 rue des Coutures, 14066 Caen, France*
[2] *Service Architecture Lab, Telecom SudParis, Institute Telecom, CNRS 5157-9 rue Charles Fourier, 91011 Evry, France*
[3] *Telecom Bretagne, Institute Telecom, Campus de Rennes, 2 rue de la Chtaigneraie, 35576 Cesson Sévigné Cedex, France*

Correspondence should be addressed to Bachir Chihani, bachir.chihani@orange.com

Received 30 March 2012; Accepted 30 May 2012

Academic Editor: MoonBae Song

Context aware communication services rely on information sources and sensors, to derive users' current situation and potential needs, and to adapt their communication services accordingly. If extensive studies have been driven on context awareness by industrials and researchers from academia, the design of such systems without modifying uses and manners of underlying communication services—while keeping them simple, intuitive, and reactive—remains a challenge. In this work, we introduce a context aware communication system that takes into account user's preferences, workload, and situation to customize telephony services. In this implementation, we use IMS for communication management. The benefits of this implementation are the enhancement of IMS with context awareness features, and the coupling of user preferences with contextual information to provide improved service customization, without modifying the user experience.

1. Introduction

In terms of usage, people still mostly use telephony like in the early 1970s; Alice dials Bob's number and hears a ring back tone, Bob picks up if he is available, and both communicate. Also, communication systems typically ignore the callee's situation when establishing a communication channel between him/her and the caller. For example the time might not be appropriate (e.g., the callee is in meeting), or the communication channel is not suitable (e.g., the callee is far from his fixed phone or is handling a mobile phone in a public space, such as a cinema).

Recently, we are witnessing big advances in devices capabilities and technology as they become smarter with the introduction of sensors (e.g., accelerometer, GPS). These advances should help to improve the legacy telephony services and to design services in adequate way with users' increasing requirements [1]. Now, effort of developing innovative communication services has to consider providing flexible mechanisms to support service adaptation based on observed changes in user situation.

Context-aware communication (CAC) [6–8] services attempt to enhance communication systems with the ability to deduce callee and caller situations in order to decrease the probability of communication failures. The situation of a user can be defined as a snapshot of the user's context at a certain instant or period of time. For a communication service, most relevant contextual information is user related (e.g., location, presence, and activity), device related (e.g., access network, battery level), and environment related (e.g., noise level).

In order to enhance existing communication services, three main challenges should be addressed.

(i) First, CAC services should extend current communication services, but without modifying the way they are used by end users, as telephony usage is very well established.

(ii) Second, CAC services should reduce the amount of interruption that current system causes and should not increase it (e.g., with inopportune notifications).

(iii) Third, CAC services should support multichannel communication (e.g., voice and text-based communications) and reactivity (e.g., not extending the call establishment delay).

Most existing work in CAC, even if some were innovative, did not address all these challenges. We intend here to tackle them by proposing a framework able to manage multiple contexts, and able to facilitate the integration of CAC services with communication services.

The rest of the paper is organized as follows. In Section 2, we present two different visions for context-awareness and we introduce the benefits and the requirements for CAC systems. In Section 3, we introduce a generic framework for supporting the development of CAC system, which is used then in Section 4 to implement CoAR (context-aware reachability), a prototype for handling incoming calls. In Section 5, we present the existing works on CAC systems, and then we discuss these works and our work in Section 6 based on the previously introduced requirements. We conclude the paper in Section 7.

2. Context-Awareness and Its Requirements

Context is all information intrinsic to an entity (e.g., user, device) that can be acquired, made explicit, and published to applications, in order to enable them to adapt their behavior to the entity's state. In this section, we present two viewpoints on context and context-awareness, typical scenarios of CAC systems, and then we present the challenges and requirements for building such innovative systems.

2.1. Theorist versus Practitioner. From the academic viewpoint [9, 10] context is about facts, rules, and axioms that can be used to describe a state of an entity (e.g., user, device) at a given time. Context-awareness is considered as a special kind of formal logic systems on which well-established artificial intelligence theories and algorithms (e.g., rule-based inference) can be applied for automating the processing of inferring new knowledge and reasoning on facts representing user situation.

From the industry viewpoint (e.g., travel or retail companies [11]) context is about preferences, activities, and geospatial information related to a given user [12, 13]. This information is acquired to answer questions like the following.

(i) Who is using the service? What are his/her preferences or habits?

(ii) What is the user doing? What are his current activities?

(iii) Where is the user? What is the logical or physical place where the service is invoked?

(iv) When? At what time are his/her actions occurring?

(v) How? Which device is used to access the service?

With this in mind, context-awareness for industries is defined as the ability of a service to adapt its response to user's requests depending on whether the service is accessed from a mobile phone or accessed from a laptop, on his/her identity and his/her role (e.g., particular or professional client), from where the user is looking for the service, and so forth.

In this paper, while keeping in mind the academic viewpoint, we intend to take into account the industry viewpoint that is usually less investigated.

2.2. Benefits of CAC Systems. In order to offer context-awareness, communication systems should be able to detect user-relevant situations and to adapt their behavior accordingly, as well as their user interface. For example, the fact that a user is working or on a holiday could impact the handling of incoming calls (e.g., to reject a call from a best friend or a call from a business partner), and even the user's contact list (how the contact list is ordered).

Also, the way received messages and event notifications are displayed should be adapted to the user situation; an urgent voicemail can be played by the TV if the user is watching TV alone, but when someone is with them then the message will be displayed on his/her smartphone. Such systems will be smart enough to take into account rules specifically related to the user environment (e.g., do not answer calls during a movie), while considering exceptions (e.g., receiving a call considered urgent by a well-known caller).

2.3. Requirements for CAC Systems. The scenarios presented above involve different equipments with different capabilities and technologies, for example, hardphone, softphone, smartphone, TV, and PC screens. They involve different sources of information (e.g., agenda, contact list, preferences) concerning the users and their surrounding environment (e.g., nearby communication devices, transportation means) to help in deducing a users' situation. To realize such scenarios, big challenges have to be faced like how to enhance existent services without having to modify their implementation and usage, how to make system as simple as possible even when deployed in complex environments, how to consider multiple communication channels, and how to support real-time interaction with the user.

To summarize, CAC systems should address the six following requirements, which can be grouped as follows.

User-related requirements:

(1) Heterogeneity of components (sensors, mobile devices, servers), communication technologies and protocols (e.g., SIP, XMPP);

(2) Ease of use because end users will not be IT experts;

(3) Privacy of users, which has to be controlled (e.g., how user's information will be used by applications) and preserved (e.g., who can see what); and

System-related requirements:

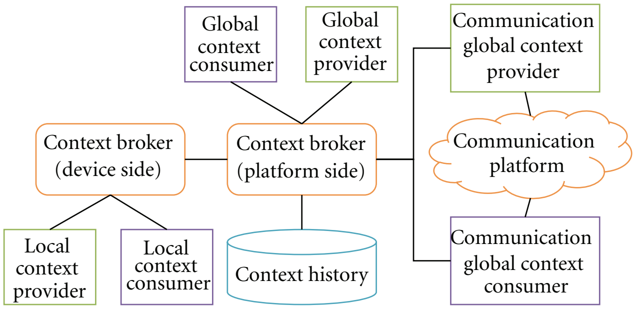


Figure 1: Context management framework.

(4) Flexibility of information modeling and management (acquisition, distribution, processing and storage);

(5) Scalability from few to many users and devices, often belonging to different administrative domains;

(6) Reactivity in order to guarantee a real time adaptation and a reasonable response time.

These requirements have to be addressed over the whole life cycle of contextual information (from the extraction to its use) and at the different parts of the CAC system.

3. Framework for Developing CAC Systems

3.1. Context Management. To develop prototypes of context-aware systems, we propose a generic framework for integrating context and communication management.

Figure 1 illustrates the architecture of our framework, which is based on the consumer/provider design pattern with the introduction of brokers to decrease the amount of point-to-point communication between the different components. In this figure, the green color is used to represent providers, the orange for brokers, the blue for databases, and the purple for consumers. The device side broker (implemented for a PC and an Android client) aims to decouple local context providers from local context customers and remote components, while the platform side broker (implemented as a Restlet application running on a Jetty web server) aims to decouple the different components and third-party applications. Local context providers/consumers are components present on a device, while global context providers/consumers are components (e.g., third-party applications) that are connected directly to the platform-side context broker. In this architecture, local brokers are able to aggregate contextual information produced by local context providers before publishing it into the global context broker to be available for all the different components in order to reduce the amount of exchanged messages between both sides. We used the CometD (http://cometd.org/) server as the notification server to implement the publish/subscribe mechanism for context distribution.

For gathering communication-related context, a specific global context provider is used, the communication global context provider. We use the communication global context consumer for performing adaptation as a way to improve communication management.

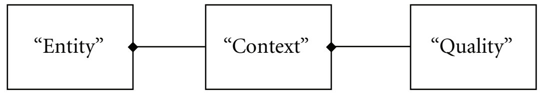


Figure 2: Context modeling approach.

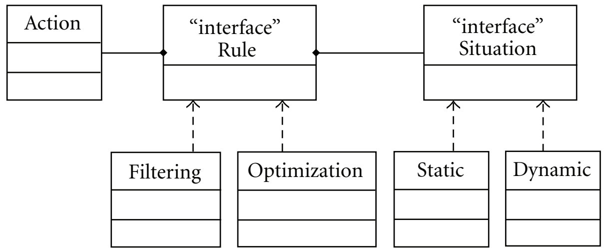


Figure 3: Rule diagram.

3.2. Context Modeling. Figure 2 illustrates the generic context model used by the framework. It is composed of three interfaces: Entity (e.g., person, place, device, network), Context (e.g., location of a person, connectivity of a device), and Quality (e.g., error rate of a sensor). The interfaces have to be implemented in order to create a complete context model. The context is transported in XML between the architecture components.

3.3. Context Reasoning. We propose a generic diagram (Figure 3) for modeling context reasoning rules. A reasoning *Rule* is an IF-THEN rule; the condition part is a set of *Situation* instances, and the action part is a set of *Action* instances. A rule can be for filtering (e.g., blocking calls when a user is in a meeting) or for optimization (e.g., forward a call to a hardphone instead of the mobile if the user is in his/her office). A situation is a set of contextual information; it can be static and set by a user before runtime (e.g., the user is in a meeting) or dynamically known at runtime (e.g., a user's current situation).

The Algorithm 1 for handling context reasoning rules takes the list of registered rules (e.g., user preferences) and the current user situation as parameters. First, filtering rules are evaluated; the situation part of each rule is evaluated against the user's current situation to retrieve the corresponding actions. Then, the resulting actions are evaluated against the user's devices' current situation in order to eliminate the nonapplicable actions (e.g., forward call to a not registered phone). Finally, the optimization rules are used to evaluate the applicable actions and to output the actions that most fit the user's current situation. They allow the framework to deal with rule collision cases, where more than an action can be performed by ranking applicable actions and eliminating those not fitting user current situation. A threshold α is used to measure the fitness (weight) of an action given user current situation; actions

Input:
Current user situation or context data $C = \{c_1, \ldots, c_n\}$,
Fitness Threshold α,
Filering rules $\mathrm{FR} = \{\mathrm{fr}_1, \ldots, \mathrm{fr}_p\}$,
Optimization rules $\mathrm{OR} = \{\mathrm{or}_1, \ldots, \mathrm{or}_q\}$

Output:
Applicable actions $A = \{a_1, \ldots, a_m\}$

A$\leftarrow \emptyset$
Applicable $\leftarrow \emptyset$
for each rule $\mathrm{fr}_i \in \mathrm{FR}$ **do**
  if $\mathrm{fr}_i \cdot$ *condition* $\in$ C **and** d $\mathrm{fr}_i \cdot$ *action*$\cdot$*is Applicable* ():
    Applicable $\leftarrow$*Applicable* $\cup$ $\mathrm{fr}_i \cdot$ *action*
for each rule $\mathrm{or}_i \in \mathrm{OR}$ **do**
  for each action $a_j \in$ *Applicable* **do**
    $a_j \cdot weight = a_j \cdot weight + \mathrm{or}_i \cdot evaluate\,(a_j)$
for action in $a_i \in$ *Applicable* **do**
  if $a_i \cdot weight > \alpha$:
    $A \leftarrow A \cup a_i$
return A

Algorithm 1: Reasoning on contextual information.

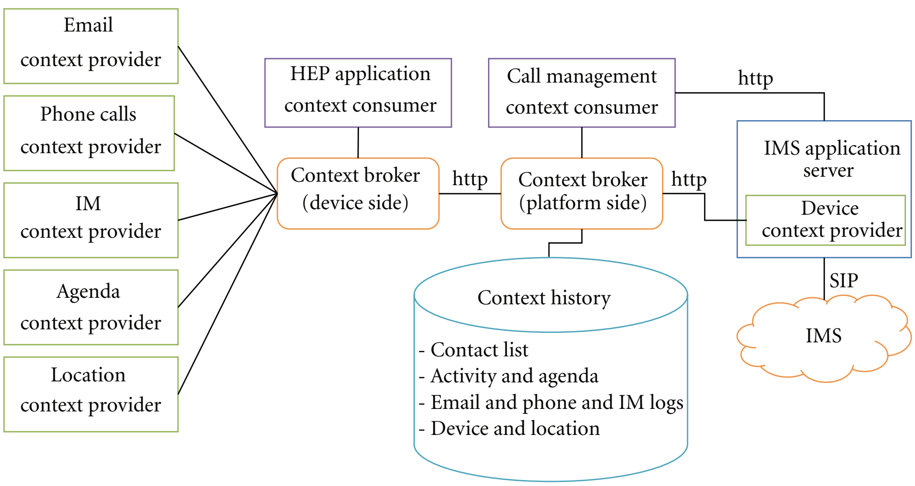


Figure 4: Architecture of the call management solution.

with low weight will be eliminated and not executed. An example of optimization rules would be a rule that prioritize the desk phone over cell phone in case both can receive the incoming call.

In our implementation, the threshold is set to zero so that any rule positively evaluated will be executed.

4. CoAR: Implementation with an IMS Platform

4.1. Architecture Overview. IMS [14] is a standardized architecture for converged communication services. It is structured into three planes: the application plane for offering end users' services through IMS application servers, the control plane for implementing access control and call/data routing, and the transport plane, which consists of physical resources for transporting voice and data.

Figure 4 illustrates the architecture of our prototype for call management. The components are as follows.

(i) A set of local context providers that periodically retrieve contextual information (e.g., the number of received/rejected/answered phone calls) from various communication tools (e.g., E-mail, IM, phone, and agenda).

(ii) A local context consumer in charge of generating the user workload level from the gathered context, in order to publish this workload so that it is accessible to users' contacts.

(iii) A device-side broker that decouples local context providers, consumers, and remote components.

(iv) A platform-side broker to facilitate the distribution of users' workload levels among the different consumers.

(v) A database for storing all the gathered context information for a future use.

(vi) A location context provider to track user's location (e.g., at office, home) and his/her transportation means (e.g., highway, train).

(vii) A device context provider to provide device-related contextual information (e.g., type, battery level, nearby devices) and IMS-related information (e.g., registration status, timestamp and expiration time, presence status). It plays the role of the communication global context provider, and it is implemented with the service triggering module provided by Mobicents.

(viii) A call management context consumer in charge of making decisions related to call routing in IMS architectures (e.g., call transfer/forwarding/completion). In the request body sent to this component, devices' IMPUs are included to identify corresponding users. This device IMPU represents an ID in REST requests.

The call management context consumer plays the role of the communication global context consumer. It is implemented as a RESTful web application. It has two interfaces: one with the context management framework for retrieving contextual information and another with the IMS AS for handling routing decision requests. The use of RESTful web services with HTTP as underlying transport protocol instead of SIP aims to make the framework generic enough to be used in other environments than SIP/IMS.

4.2. Use Case: Handling of Incoming Calls. When the IMS receives a call request (i.e., a SIP Invite) for a given IMPU (IMS public user identity, that is, the identifier that is used to reach a user on corresponding device), the Mobicents AS is triggered. The AS sends a request to the call management context consumer with the IMPUs of both caller and callee. It receives in response the action to perform on this call request. The different actions are as follows:

(i) nothing: to let the call reach the callee;

(ii) busy: to interrupt the incoming call;

(iii) redirect: to redirect the call to a voice server; or

(iv) transfer: to transfer the call to another destination.

In the redirect and transfer cases, an optional parameter specifies the target IMPU (e.g., IMPU of the voice server). If the call management context consumer module does not respond in due time, the "nothing" action is performed.

When reasoning based on contextual information and deciding how to handle incoming calls based on the user situation, it is very important to include user preferences. For example, users may not want to be interrupted by a call from a certain class of contacts (e.g., colleague) if they are very focused on their current work (high workload level). In this implementation we used the previously described rule diagram (Figure 3) to represent user preference rules.

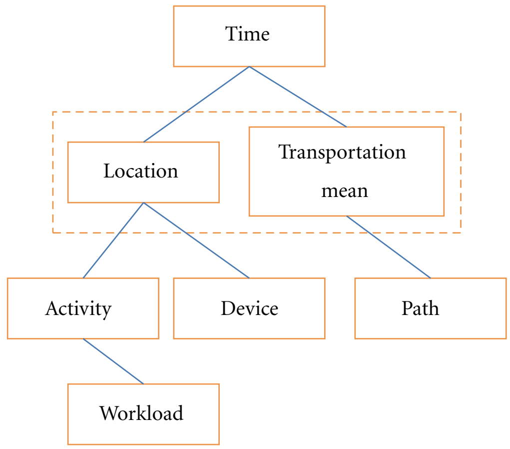


Figure 5: Hierarchical view of contextual information.

Concretely, these rules are represented in XML so that they can be exchanged easily between framework components; they are instantiated as java objects to be processed by the internal reasoning engine. We are using a custom rule engine that consumes java objects representing filtering or optimization rules to generate the action to be performed.

To help users defining their preference rules and to guide them through this process, we propose a hierarchical view (Figure 5) for representing contextual information to users. For a preference rule, users may first indicate the type of time (e.g., workday, weekend, vacation, lunch time). Then, they have to choose a location (e.g., office, home) or the transportation means and type of path (e.g., home-work path, occasional path) if the rule will be applied when a user will be on the move. Next, they can specify the type of activity (e.g., reading mails) and device (e.g., hardphone) affected by this rule. For the activity, they can specify the workload level (e.g., busy). Following this approach, an example of a generated rule will be as Algorithm 2.

CoAR here allows users to have control over their communication. Users specify their preferences regarding how they can be reached under which situation. It provides an extension to existent communication platform without modifying underlying communication service and reduces amount of interruptions caused by existing communication service as it decides on how calls are managed on behalf of user. Also, it considers context information from multiple sources.

4.3. Infrastructure and Evaluation. Our testbed IMS platform is built on the Open IMS Core [15] which implements the control plane of IMS architecture (P-CSCF, I-CSCF, S-CSCF, HSS). At the application plane, we used Mobicents [16] application server (AS) for communication service creation, through its JAIN SLEE API. We use also different types of IMS clients (e.g., IMSDroid for Android-based smartphones, Uct IMS client for softphones).

> **Filtering rule:**
>
> If isTime (Workday) == true
>   and isLocation (AtWork) == true
>   and isActivity (EditDocument) == true
>   and isWorkload (Do_Not_Disturb) == true
>   and isCallFrom (Boss) == false
> Then
>   RedirectCallToVoiceServer (VoiceXML_File)

Algorithm 2

Figure 6: Test environment.

Figure 6 depicts our test environment configuration. The Open IMS Core is installed on Ubuntu/Linux 10.0 operating system running on Oracle VirtualBox (https://www.virtualbox.org/) virtual machine. The latter, as well as Mobicents AS, is running on Microsoft Windows XP SP3 installed on Dell D630 laptop with the following characteristics vender: Intel, model: Core 2 Duo, CPU: 2,2 GHZ, memory: 2 GB.

For persistence, contextual information are stored on an object database DB4O (http://www.db4o.com/). Context Broker and call management context consumer are both implemented as web applications with the Restlet (http://www.restlet.org/) framework running on a Jetty web server. Both are installed on Microsoft Windows 2003 Server SP2 with the following hardware charcteristics: vender: Intel, model: Xeon, CPU: 2,8 GHZ, memory: 2 GB.

Main contribution of this paper is the design and implementation of a context-aware communication management framework able to provide user with support and assistance service by handling his/her preferences and context parameters. Therefore, it is important to evaluate the cost to achieve context-awareness in terms of response time as it is an additional amount of time to the response time of the communication service. The summation should remain acceptable for users.

The response time of CoAR depends on type of operation (context publication, context query/consumption, or context subscription), and needed time to upload contextual information from database to memory in case it is stored. The response time for the call management context consumer depends on response time of the broker plus the service reasoning time. User response time depends on response time of Mobicents AS that depends on call management context consumer response time.

The following graphs illustrate the evolution of response time for broker according to the number of parallel context publications (Figure 7) and number of parallel context consumption requests (Figure 8).

The aim is to figure out the response time as function of these parameters in order to measure the reactivity of the overall system and to be able to estimate in future the expected response time.

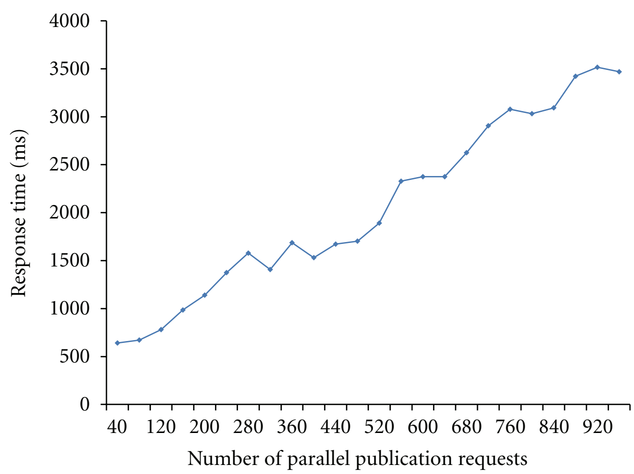


Figure 7: Response time variation according to number of publication requests.

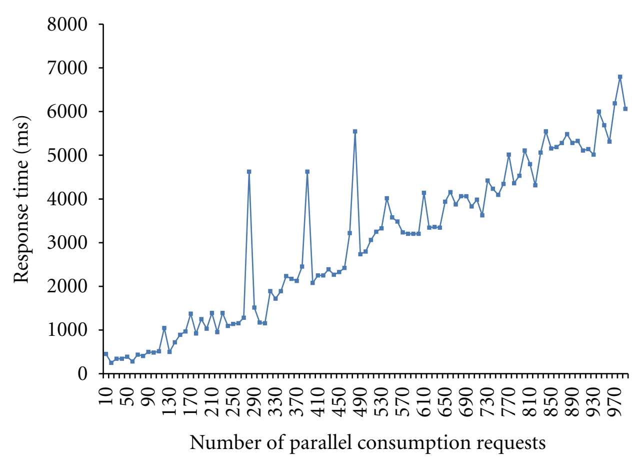


Figure 8: Response time variation according to number of consumption requests .

The two above graphs present the variation of response time of parallel context publication/consumption requests. Both match the following linear function:

$$\text{Response Time (Request Number)} = \alpha + \beta * \text{Request Number} \tag{1}$$

The function parameters α, β can be approximately computed as follows:

$$\beta = \text{average}\left(\frac{\Delta\text{Response Time}_{i+1,i}}{\Delta\text{Request Number}_{i+1,i}}\right), \tag{2}$$

$$\alpha = \text{average}\left(\text{Response Time}_i - \beta * \text{Request Number}_i\right). \tag{3}$$

For the first graph data we find $\beta \approx 3{,}07$ and $\alpha \approx 516{,}46$, for second graph we find $\beta \approx 5{,}72$ and $\alpha \approx 143{,}47$.

We believe that the current CoAR response time should be acceptable even under extreme conditions, that is, when hundred of requests are received in the same time. To enhance it further, we should investigate IO operations (input/output) response time derived by storage and retrieval of contextual information from the database, for example, by using cache systems.

The delay time engendered by our framework before call establishment between two users depends on needed time for sending an HTTP request to the call management service and the reasoning time needed to decide how an incoming call should be handled. Figure 9 depicts a comparative summary of delay distribution in case user preferences are used to decide how to handle incoming calls (i.e., with), or not used (i.e., without). The distributions are generated by sending about one hundred sample requests to the call management context consumer component. In the figure, the median (50th percentile) of each distribution is represented with a darkened line; bottom side of boxes represents the 25th percentile of corresponding distribution; top side represents 75th percentile; bottom line represent 10th percentile; top line represents 90th percentile.

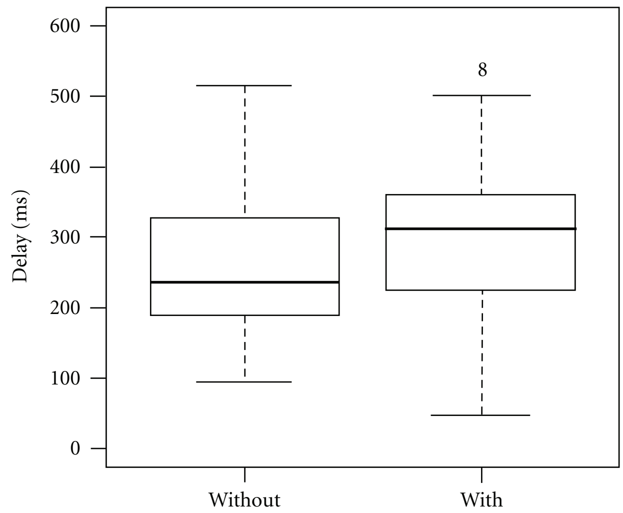


Figure 9: Distribution of call establishment delay.

From the figure, the two distributions are very close in terms of shape, similar dispersion (spread) of delay samples. Distribution median is about 300 ms when reasoning on user preferences and about 240 ms when no reasoning is performed, this difference is due to reasoning overhead. All values of both distributions are between 50 ms to 500 ms, with an interquartile (samples in the box between its bottom and top) ranging from 200 ms to 350 ms. As a result, the delay is more influenced by time needed to send HTTP request to the call management service and reasoning overhead is negligible against it. This delay range is fully compatible with the duration of an end-to-end call setup.

5. Related Work

A first experiment with communications management has already been successfully conducted in [17, 18] where we defined HEP (enHanced unintErruPtibility) which provides users with the workload level of their contacts on different communication channels to help them to evaluate the impact of the interruption they will cause by initiating a communication with a given contact. The system was efficient in representing the workload of contacts, but in some cases users ignored the provided information and still initiated communications with the callee. We believe that a communication management system only based on publicizing information about the availability of callees is not sufficient. However, communication management systems based only on routing policies entered by the callee, such as call transfer, suffer from a lack of flexibility. Hence, we propose CoAR a communication management system that merges a context management framework with a communication platform in order to best manage users' communications.

Cals.Calm [19] provides information about callees by publishing them on a web page. Thus any device with Internet connectivity can access them, enabling the support for heterogeneity but also presenting a serious threat to callees' privacy. Considered context is limited to location and activities which are manually edited by users, and the system does not provide any intelligence to make use of this information. The system is based on a web server that may be configured to support scalability and real time responsiveness.

MOCSP [20] defines communication ext-links, URI, to a web-based communication service (e.g., IM, E-mail, etc.), as the address to be used by a caller to reach users. In [21], MOCSP is used for managing communication sessions across callee's multiple devices based on user preferences. Users have to specify which communication service to be used for a group of contacts. The system is open to the Internet and users may access it with any device having Internet connection. The system is based on a P2P model in which each user has its own communication server.

In [2] a CPL- (Call-Processing-Language-) based CAC for customizing call control services (forward, transfer, and completion) is presented. Decisions rely on user-entered preferences which may lead to incorrect call filtering decisions and/or threaten user privacy. The system considers users as having the capability to build CPL scripts and provides a CPL scripting editor and a command line for configuration. Contextual information is limited to location and calendar entries. System architecture is client-server, providing a means for support scalability.

In [3] a centralized system based on otology approach for context modeling and reasoning to adapt the way incoming calls (e.g., not answered) are handled (e.g., forwarding) is presented. Considered context is historical information of communication sessions already established between the callee and the caller without exposure to third parties, that is, no support for heterogeneity. Routing decisions rely on a common threshold to all users, making it inflexible and not able to adapt to quick variations in a user's situation.

AmbiTalk [22] is an SIP-based CAC system to enable smart places to enforce their communication policies on mobile devices, which limits supported devices. It allows mobile devices to adapt their configuration (e.g., turn phone to vibrating mode when it enters a meeting room) and to negotiate communication sessions (e.g., IM instead of phone call if two devices are located within places enforcing different policies). Thus, it may restrict device usage and raise usability and privacy concerns. Considered context is limited to device-related (e.g., location, alert mode, or owner).

In [4] a centralized policy-based context-aware message delivery system is presented. The system enhances OMA (Open Mobile Alliance) Converged IP Messaging (CPM) with context-awareness for selecting appropriate messaging services (e.g., SMS, E-mail). User preferences are used in the condition portion of IF-THEN routing policies which may lead to wrong decision, that is, routing a message to a bad destination. It requires some user configuration efforts to specify for each contact the preferred communication channel.

INCA [23] is a layered CAC assistant with a P2P-based architecture enabling scalability. Two consecutive layers interact with event notifications and requests for action execution, which may lead to conflicts between actions made by different layers and/or to wrong decisions. Contextual information are limited to user preferences, used devices, type of communication technology (e.g., VOIP), and subject of the communication.

SECE [24] (Sense Everything, Control Everything) is a context management framework supporting the creation of mashable communication services that combine multiple context dimensions (location, presence, agenda). The framework enables end users to write context-aware Tcl-based scripts composed of conditions on user context and corresponding actions which are described in a simple and natural language terminology. Even if writing context-aware rules (e.g., send a happy birthday message on behalf of user when it is the birthday of a friend) is intuitive, it is hard for users to maintain and keep such rules up to date, especially for actions that are not frequent in a uniform way or may happen under ambiguous situations.

In [5], the authors present an approach for embedding device context into mobile search queries to enhance the relevance of the result to device characteristics and user situation. They propose a rule-based approach for abstracting raw physical measurements (e.g., GPS location) acquired from the device sensor into user high-level contextual information (e.g., user is at home). These derived information are then transferred to search engines (e.g., Bing) as additional parameter using the corresponding API which may limit the potential of the solution as it depends on a limited set of acceptable parameters for the search engine.

6. Discussion

Most of the works presented in the previous section allow users to define rules that specify how calls have to be routed, or provide caller with information related to callee's situation to make decision on whether or not making a call. The first option lacks flexibility, and is hardly maintainable as users may not update their rules every time their preferences change. The second option assumes willingness of caller to use the provided information to make better decisions and avoid interrupting callee. Another option would be to automatically detect user context and adapt service behavior correspondingly as proposed by CoAR.

We summarize the discussion of the different works in Table 1 where evaluation criteria are based on the requirements already introduced in Section 2.2. The symbol "+" means the existence of a support for the corresponding requirement, whereas the symbol "−" indicates its absence; "±" is used to describe cases where support exists but needs to be enhanced.

Based on this analysis, we can conclude that most of the previous work in developing CAC systems has been focusing more on enhancing the system-related requirements rather than user-related requirements. In our work, we try to provide a better means to meet both system- and user-related requirements.

Our framework (Figure 1) supports the enhancement of communication platform generally and IMS in particular with context-awareness features for enabling service customization. Yet, it does not require changes in user habits regarding the use of their favorite communication services, and it reduces interruptions caused by underlying services. Also, it is generic enough to be used in other environments than SIP/IMS as it is based on RESTful web service with HTTP as underlying transport protocol and not SIP.

The main difference between our model (Figure 2) and the different context models that have been proposed is its simplicity that enables developers to use the framework without having to understand complex modeling schema, thereby making it possible to rapidly prototype context-aware systems.

The distribution of brokers makes the system scalable and provides a better support for heterogeneity because a device-side broker can be implemented on different devices with different technologies. Implementing lightweight providers/consumers on the device side provides an enhanced privacy protection because contextual information can be used locally; they can also be carefully designed to provide an enhanced usability. Implementing

Table 1: Comparison between CAC systems.

	(1) Heterogeneity	(2) Usability	(3) Privacy	(4) Flexibility	(5) Scalability	(6) Reactivity
Cals.Calm	+	−	−	−	+	+
MOCSP	++	±	−	−	++	+
Framework [2]	+	−	−	+	+	+
Framework [3]	−	−	−	−	−	−
AmbiTalk	−	−	±	−	+	+
Framework [4]	+	−	−	−	−	−
INCA	±	−	−	+	++	+
SECE	++	+	−	±	+	+
Framework [5]	±	+	+	±	+	+
CoAR	+	±	±	+	±	±

providers/consumers on the platform side and the simplicity of the context modeling approach enable the use of any contextual information sensed or derived from running data mining algorithms.

Our framework provides a support for reasoning on context and system adaptation via the Rule class diagram (Figure 3) which is generic enough to represent any kind of adaptation rules. In addition, the reasoning engine is able to interpret two types of rules (i.e., filtering and optimization) to produce a more flexible adaptation. This distinction between rules is a logical distinction as filtering rules implement hard constraints on actions (if condition is not verified then eliminate corresponding action) while optimization rules implement soft constraints for evaluating and ranking actions. This distinction motivated the choice of a custom rule engine instead of using a generic rule engine.

For the moment, the framework does not support cases where multiple devices use the same SIP ID/IMPU. To address this limitation, contact header in SIP messages can be used to store more information (e.g., device location, IP address) that help distinguish between multiple devices with same IMPU. For improving performance of reasoning, we should use more sophisticated engines able to process adaptation policies described in a domain-specific language.

7. Lessons Learnt and Perspectives

In this work, we present a framework that aims to manage contextual information on behalf of applications and provide them with mechanisms for context-awareness that does not imply any changes to the original application logic and the interaction between user and the application. Three main lessons were learned from our work.

(i) First, the IMS is sufficiently flexible to support an external call routing decision based on the context of the callee.

(ii) Second, http quasi-synchronous mechanisms like CometD enable to develop nearly real-time publish/subscribe RESTful services to distribute the context among the providers and consumers.

(iii) Third, a cache system seems more adequate than a database for context storage, as context information has a limited validity time. This leads also to better performances.

More generally, this work reveals the limitation of rule-based contextualization approaches as they are not flexible enough due to the need for maintaining continuously the rule base. It also shows the importance of user feedback in the process of customizing adaptation. This feedback can be acquired directly from user (explicit feedback) or by observing user behavior and reactions (implicit feedback).

An interesting perspective would be to use contextualization approaches that take as input user feedbacks (implicit and explicit) in addition to context data in order to generate and update autonomously inference rules. Machine learning approaches are promising for this purpose and should be deeply investigated. These approaches will enable automated reasoning on contextual information without need for predefining and maintaining inference rules.

In the future, we plan to make the decision process more flexible by supporting the automated learning of routing rules. We intend also to interface our framework with an IPTV system so that we can implement the second part of the presented use cases.

8. Conclusion

Communication services extended with end users context awareness features will help reduce and in some case eliminate barriers that complicate communication between individuals and groups.

This paper introduces a framework that integrates a communication platform with a comprehensive context management system. It describes how this framework was used to prototype CoAR, a CAC system that combines contextual information with user preferences to decide how to handle incoming calls. We believe that our work illustrates the possibility of building innovative CAC systems that rely on existing infrastructure without disturbing their usage or their performances.

What still need to be investigated by researchers from industrial companies and academic universities is how to gain user's trust and acceptance toward context-aware systems, as the context-awareness benefits do not come without cost, especially the threats on user privacy implied by the system autonomous monitoring of context and activities in a daily manner.

References

[1] D. Tacconi, D. Miorandi, I. Carreras, F. De Pellegrini, and I. Chlamtac, "Cooperative evolution of services in ubiquitous computing environments," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 6, no. 3, article 20, 2011.

[2] M. Görtz, R. Ackermann, J. Schmitt, and R. Steinmetz, "Context-aware communication services: a framework for building enhanced IP telephony services," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN'04)*, pp. 535–540, Chicago, Ill, USA, October 2004.

[3] K. Hamadache, E. Bertin, A. Bouchacourt, and I. Benyahia, "Context-aware communication services: an ontology based approach," in *Proceedings of the 2nd International Conference on Digital Information Management (ICDIM'07)*, Lyon, France, October 2007.

[4] N. Blum, S. Lampe, and T. Magedanz, "Enabling context-sensitive communication experiences," in *Proceedings of the 14th International Conference on Intelligence in Next Generation Networks (ICIN'10)*, Berlin, Germany, October 2010.

[5] E. Yndurain, D. Bernhardt, and C. Campo, "Augmenting mobile search engines to leverage context awareness," *IEEE Internet Computing*, vol. 16, no. 2, pp. 17–25, 2012.

[6] B. N. Schilit, D. M. Hilbert, and J. Trevor, "Context-aware communication," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 46–54, 2002.

[7] A. Ranganathan and H. Lei, "Context-aware communication," *Computer*, vol. 36, no. 4, pp. 90–92, 2003.

[8] F. Toutain, A. Bouabdallah, R. Zemek, and C. Daloz, "Interpersonal context-aware communication services," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 68–74, 2011.

[9] P. Mehra, "Context-aware computing: beyond search and location-based services," *IEEE Internet Computing*, vol. 16, no. 2, 2012.

[10] J.-D. Kim, J. Son, and D.-K. Baik, "CA_{5W1H} onto: ontological context-aware model based on 5W1H," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 247346, 11 pages, 2012.

[11] W. Clark, *How Context Can Improve Your Customer Relationships*, Gartner Webinars, 2010.

[12] Z. Zhao, N. Lage, and N. Crespi, "User-centric service selection, integration and management through daily events," in *Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM'11)*, pp. 94–99, Seattle, Wash, USA, March 2011.

[13] R. Tusch, M. Jakab, J. Köpke et al., "Context-aware UPnP-AV services for adaptive home multimedia systems," *International Journal of Digital Multimedia Broadcasting*, vol. 2008, Article ID 835438, 12 pages, 2008.

[14] R. Copeland, *Converging NGN wireline and mobile 3G networks with IMS: converging NGN and 3G mobile*, CRC Press, 2008.

[15] D. Vingarzan, P. Weik, and T. Magedanz, "Development of an open source IMS core for emerging IMS test-beds, academia and beyond," *Journal for Mobile Multimedia*, vol. 3, no. 2, pp. 131–149, 2006.

[16] A. Bhayani, "Developing converged application using open source software," in *Proceedings of the IEEE Symposium on Industrial Electronics and Applications, ISIEA 2009*, pp. 452–456, October 2009.

[17] B. Chihani, E. Bertin, and N. Crespi, "A comprehensive framework for context-aware communication services," in *Proceedings of the 15th International Conference on Intelligence in Next Generation Networks (ICIN'11)*, Berlin, Germany, October 2011.

[18] B. Chihani, E. Bertin, F. Jeanne, and N. Crespi, "Context-aware systems: a case study," in *Proceedings of International Conference on Digital Information and Communication Technology and its Applications (DICTAP'11)*, Dijon, France, 2011.

[19] E. R. Pedersen, "Calls.calm: enabling caller and callee to collaborate," in *Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI'01)*, pp. 235–236, New York, NY, USA, 2001.

[20] S. Shanmugalingam, N. Crespi, and P. Labrogere, "My own communication service provider," in *Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT'10)*, pp. 260–266, Moscow, Russia, October 2010.

[21] S. Shanmugalingam, N. Crespi, and P. Labrogere, "User mobility in a web-based communication system," in *Proceedings of the IEEE 4th International Conference on Internet Multimedia Services Architecture and Application (IMSAA'10)*, Bangalore, India, December 2010.

[22] E. Karmouch, A. Nayak, P. Martin, and H. Hassanein, "Experimenting with mobile context-aware sip-based multimedia communications," in *Proceedings of the 1st International Global Information Infrastructure Symposium (GIIS'07)*, pp. 6–13, Marrakech, Morocco, July 2007.

[23] B. E. Saghir and N. Crespi, "A generic layer model for context-aware communication adaptation," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'08)*, pp. 3027–3032, Las Vegas, Nev, USA, April 2008.

[24] O. Boyaci, V. Beltran, and H. Schulzrinne, "Bridging communications and the physical world," *IEEE Internet Computing*, vol. 16, no. 2, pp. 35–43, 2012.